Smart Contract Audit Report

Dolomite

Audit Performed By

Fortknox Security

Professional Smart Contract Auditing

January 11, 2024

Table of Contents

Executive Summary

Fortknox Security has conducted a comprehensive smart contract security audit for **Dolomite**. Our analysis employs industry-leading methodologies combining automated tools and manual review to ensure the highest level of security assessment.

23	6	MEDIUM	100%
TOTAL ISSUES FOUND	CRITICAL + HIGH	OVERALL RISK	CODE COVERAGE

Security Assessment Overview

Critical Issues

2

Immediate action required. These vulnerabilities can lead to direct loss of funds.

IMPACT: SEVERE FINANCIAL LOSS

High Issues

4

High priority fixes needed. Can lead to significant financial loss.

IMPACT: MAJOR SECURITY RISK

Key Findings Summary

Access Control

Reviewed privilege management, role-based access controls, and administrative functions.

Economic Security

Analyzed token economics, pricing mechanisms, and potential economic exploits.

Logic Validation

Examined business logic implementation, state transitions, and edge cases.

Input Validation

Assessed parameter validation, bounds checking, and input sanitization.

Audit Conclusion

The Dolomite smart contract audit reveals **23 total findings** across various security categories. Immediate attention is required for 6 critical/high severity issues before deployment. Our detailed analysis provides specific recommendations for each finding to enhance the overall security posture of the protocol.

Audit Methodology

Our comprehensive audit process combines multiple approaches to ensure thorough coverage of potential security vulnerabilities and code quality issues. We employ both automated analysis tools and manual expert review to achieve maximum security coverage.

Tools & Techniques

Static Analysis

Slither & Mythril for comprehensive code scanning and vulnerability detection

Manual Review

Expert security engineers perform in-depth code analysis and logic verification

Business Logic

Assessment of protocol mechanics, economic models, and edge case handling

Gas Analysis

Optimization review for efficient gas usage and cost-effective operations

Formal Verification

Mathematical proof methods to verify critical contract properties

Symbolic Execution

Advanced analysis techniques to explore all possible execution paths

Review Process & Standards

Review Process

1 Initial Scanning

Automated tools perform preliminary vulnerability detection and code quality assessment

2 Manual Review

Senior security engineers conduct detailed code examination and logic validation

3 Business Logic Testing

Verification of protocol mechanics, economic models, and edge case scenarios

4 Architecture Analysis

Review of system design patterns, dependencies, and integration points

5 Final Documentation

Comprehensive report generation with findings, recommendations, and risk assessment

Severity Classification

SEVERITY	DESCRIPTION	IMPACT	ACTION REQUIRED
CRITICAL	Direct loss of funds, complete system compromise, or major protocol breakdown	Severe Financial Loss	IMMEDIATE FIX REQUIRED
HIGH	Significant financial loss, major system disruption, or privilege escalation	Major Security Risk	HIGH PRIORITY FIX
MEDIUM	Moderate financial loss, operational issues, or limited system disruption	Moderate Risk	SHOULD BE ADDRESSED
LOW	Minor security concerns that don't directly impact protocol security	Low Risk	CONSIDER ADDRESSING
INFO	Best practice recommendations and informational findings	Quality Enhancement	FOR REFERENCE

Audit Scope

Project Details

PARAMETER	DETAILS
Project Name	Dolomite
Total Issues Found	23
Audit Type	Smart Contract Security Audit
Methodology	Manual Review + Automated Analysis

Files in Scope

This audit covers the smart contract codebase and associated components for Dolomite.

Audit Timeline

- ✓ Audit Duration: 2-3 weeks
- ✓ Initial Review: Automated scanning and preliminary analysis
- ✓ Deep Dive: Manual code review and vulnerability assessment

Vulnerability Analysis

Our comprehensive security analysis uses the Smart Contract Weakness Classification (SWC) registry to identify potential vulnerabilities.

SWC Security Checks

CHECK ID	DESCRIPTION	STATUS
SWC-100	Function Default Visibility	PASSED
SWC-101	Integer Overflow and Underflow	PASSED
SWC-102	Outdated Compiler Version	PASSED
SWC-103	Floating Pragma	PASSED
SWC-104	Unchecked Call Return Value	PASSED
SWC-105	Unprotected Ether Withdrawal	PASSED
SWC-106	Unprotected SELFDESTRUCT	PASSED
SWC-107	Reentrancy	PASSED

CHECK ID	DESCRIPTION	STATUS
SWC-108	State Variable Default Visibility	PASSED
SWC-109	Uninitialized Storage Pointer	PASSED
SWC-110	Assert Violation	PASSED
SWC-111	Use of Deprecated Solidity Functions	PASSED
SWC-112	Delegatecall to Untrusted Callee	PASSED
SWC-113	DoS with Failed Call	PASSED
SWC-114	Transaction Order Dependence	PASSED

Contract Privileges Analysis

Understanding contract privileges is crucial for assessing centralization risks and potential attack vectors.

Common Privilege Categories

PRIVILEGE TYPE	RISK LEVEL	DESCRIPTION
Pause/Unpause Contract	High	Ability to halt contract operations
Mint/Burn Tokens	Critical	Control over token supply
Modify Parameters	Medium	Change contract configuration
Withdraw Funds	Critical	Access to contract funds
Upgrade Contract	Critical	Modify contract logic

Mitigation Strategies

- ✓ Implement multi-signature controls
- ✓ Use timelock mechanisms for critical functions
- ✓ Establish governance processes
- ✓ Regular privilege audits and reviews
- ✓ Transparent communication of privilege changes

C-0 | DoS Callbacks Through Simple Transfer

CATEGORY	SEVERITY	LOCATION	STATUS
DoS	CRITICAL	Global	Resolved

Description

When the callback `afterWithdrawalExecution` is triggered, it is checked that the amount of GM tokens sent to GMX to be redeemed match the amount of GM actually redeemed:

```
afterWithdrawalExecution
```

Recommendation

Resolution

Pending resolution.

C-1 | Liquidations Prevented With Pending Action

CATEGORY	SEVERITY	LOCATION	STATUS
DoS	CRITICAL	IsolationModeTokenVaultV1WithFreezable.sol: 670	Resolved

Description

When performing a liquidation, the amount the user is to be liquidated for is validated with function `_validateWithdrawalAmountForUnwrapping`:

```
_validateWithdrawalAmountForUnwrapping
```

Recommendation

Furthermore, consider restricting the `minOutputAmount` a liquidator can pass to prevent liquidation delay through self-liquidation.

```
minOutputAmount
```

Resolution

Pending resolution.

H-0 | Withdrawals Apply _minOutputAmount To One Side

CATEGORY	SEVERITY	LOCATION	STATUS
Logical Error	HIGH	GmxV2Library.sol: 151, 152	Resolved

Description

In the `executeInitiateUnwrapping` function the `withdrawalParams` are created with a `_minOutputAmount` that can only apply to either the `shortToken` output or the `longToken` output.

```
executeInitiateUnwrapping
withdrawalParams
_minOutputAmount
shortToken
longToken
```

Recommendation

Refactor the _minOutputAmount logic such that the minimum output can be split amongst the minLongTokenAmount and `minShortTokenAmount`.

```
minShortTokenAmount
```

Resolution

Pending resolution.

H-1 | Redemptions Incorrectly Appear Unpaused

CATEGORY	SEVERITY	LOCATION	STATUS
Logical Error	HIGH	GmxV2Library.sol: 236 - 239	Resolved

Description

The function `isExternalRedemptionPaused` is used to determine if GM withdrawals are currently paused, utilizing the current PnL-to-Pool Factors as one validation for a paused state.

```
isExternalRedemptionPaused
```

Recommendation

Resolution

Pending resolution.

H-2 | Severely Undercollateralized Positions Cannot Be Liquidated

CATEGORY	SEVERITY	LOCATION	STATUS
Logical Error	HIGH	AsyncIsolationModeUnwrapperTraderImpl.sol: 184-198	Resolved

Description

During liquidation, an issue appears when the total available user amount to liquidate is equal to the input liquidation amount. The input liquidation amount is passed to the `liquidate` function from the `LiquidatorProxyV4WithGenericTrader` contract when liquidating a position.

```
liquidate
LiquidatorProxyV4WithGenericTrader
```

Recommendation

In the `createActionsForUnwrapping` function from `AsyncIsolationModeUnwrapperTraderImpl` do not create a second call and sell action if the difference between the input amount and available amount is zero.

```
createActionsForUnwrapping
AsyncIsolationModeUnwrapperTraderImpl
```

Resolution

Pending resolution.

H-3 | Withdrawal Keys Misused by Differing Subaccount in Liquidations

CATEGORY	SEVERITY	LOCATION	STATUS
DoS	HIGH	AsyncIsolationModeUnwrapperTraderImpl.sol: 133-145	Resolved

Description

When a liquidation is executed on an account from a vault with multiple accounts, a malicious actor can pass the withdrawal key belonging to an account that is different from the one being liquidated and block the vault.

Recommendation

In the `_callFunction` function, verify that the stored `accountNumber` matches the `_accountInfo.number` but only if the call action was sent from a liquidation operation.

```
_callFunction
accountNumber
_accountInfo.number
```

Resolution

Pending resolution.

M-0 | Vault Owner Can Cancel Liquidation

CATEGORY	SEVERITY	LOCATION	STATUS
Access Control	MEDIUM	GmxV2IsolationModeTokenVaultV1.sol: 91	Resolved

Description

In the `cancelWithdrawal` function, the vault owner can cancel liquidations which is essentially a withdrawal of the GM tokens. When a liquidator uses `prepareForLiquidation`, they trigger a forced withdrawal from the underwater vault.

```
cancelWithdrawal
prepareForLiquidation
```

Recommendation

Differentiate between normal withdrawals and those from liquidation. Restrict the vault owner from calling the `cancelWithdrawal` function when there is a pending withdrawal initiated by the `prepareForLiquidation` function.

```
cancelWithdrawal
prepareForLiquidation
```

Resolution

Pending resolution.

M-1 | Withdrawal Not Necessarily 50-50

CATEGORY	SEVERITY	LOCATION	STATUS
Logical Error	MEDIUM	GmxV2MarketTokenPriceOracle.sol: 167-168	Resolved

Description

When calculating the swap price impact, it is assumed that withdrawing GM tokens provides 50% in the short token and the other 50% in long tokens. According to documentation, "Assume under the worst case, we liquidate 10% of the supply cap (which would entail a swap for half of that, 5%, to USDC (short token))."

Recommendation

Consider using the reader to get the current token ratios in the market and adjust the `wethAmountIn` accordingly.

```
wethAmountIn
```

Resolution

Dolomite Team: We have decided to no longer measure price impact but increase the liquidation penalty instead.

M-2 | Excess GM Not Partially Deposited On Supply Cap

CATEGORY	SEVERITY	LOCATION	STATUS
Logical Error	MEDIUM	UpgradeableAsyncIsolationModeWrapperTrader.sol: 415-421	Resolved

Description

After a deposit is created, any extra GM tokens that are received are then deposited into Dolomite Margin for the user. If this amount would equal or surpass the maximum allowed value for a market, then it is entirely sent to the vault owner.

Recommendation

Modify the `_depositIntoDefaultPositionAndClearDeposit` function so that it sends only the excess that would not fit into the market to the vault owner and deposit the rest into Dolomite in the user's account.

```
_depositIntoDefaultPositionAndClearDeposit
```

Resolution

Pending resolution.

M-3 | Can't Unfreeze Vault If Execution Interrupted

CATEGORY	SEVERITY	LOCATION	STATUS
DoS	MEDIUM	GmxV2IsolationModeUnwrapperTraderV2.sol: 124	Resolved

Description

During the unwrapping process, the vault is frozen by incrementing the mapping `_vaultToPendingAmountWeiMap` by `_amountDeltaWei.value`.

```
_vaultToPendingAmountWeiMap
_amountDeltaWei.value
```

Recommendation

Enable the Admin to invoke the `setVaultAccountPendingAmountForFrozenStatus` function, providing a means to unfreeze an account if execution is ever interrupted.

```
setVaultAccountPendingAmountForFrozenStatus
```

Resolution

Pending resolution.

M-4 | Lack Of Liquidation Incentives

CATEGORY	SEVERITY	LOCATION	STATUS
Incentives	MEDIUM	Global	Resolved

Description

According to the Dolomite whitepaper, “Liquidations forcefully repay any debt that is owed by a borrower by transferring an equivalent amount of collateral from the borrower to the liquidator, plus a liquidation penalty of 5%.”

Recommendation

Consider providing the liquidator a reward in the output token for the liquidation.

Resolution

Pending resolution.

M-5 | Withdrawals Fail When A Backing Token is Zero

CATEGORY	SEVERITY	LOCATION	STATUS
Logical Error	MEDIUM	GmxV2Library.sol: 295-300	Resolved

Description

The `outputToken` and `secondaryOutputToken` are always validated to be equal after the execution of a withdrawal:

```
outputToken
secondaryOutputToken
```

Recommendation

Modify the check such that if the value of the withdrawal output amount is 0, then the `_outputTokenAddress` and the `_secondaryOutputTokenAddress` do not have to match.

```
_outputTokenAddress
_secondaryOutputTokenAddress
```

Resolution

Pending resolution.

M-6 | Fund Transfer From Wrapper Trader Can Be Skipped

CATEGORY	SEVERITY	LOCATION	STATUS
Logical Error	MEDIUM	UpgradeableAsyncIsolationModeWrapperTrader.sol: 351	Resolved

Description

If a user receives more GM than the `minOutputAmount` they set, the excess GM has to be deposited into the Core protocol such that the Vault balance and Core balance align. However, this state assumes `_shouldSkipTransfer` has been set to `false` upon deposit creation in the call to function `IsolationModeTokenVaultV1WithFreezable.executeDepositIntoVault` :

```
minOutputAmount
_shouldSkipTransfer
false
IsolationModeTokenVaultV1WithFreezable.executeDepositIntoVault
```

Recommendation

Prior to calling `factory.depositIntoDolomiteMarginFromTokenConverter`, explicitly `_setShouldVaultSkipTransfer(/* _shouldSkipTransfer = */ false);`

```
factory.depositIntoDolomiteMarginFromTokenConverter
_setShouldVaultSkipTransfer(/* _shouldSkipTransfer = */ false);
```

Resolution

Pending resolution.

L-0 | Features May Be Disabled

CATEGORY	SEVERITY	LOCATION	STATUS
Logical Error	LOW	Global	Resolved

Description

GMX may choose to disable certain features of its protocol, including but not limited to deposit execution. If this feature was paused, a Dolomite user would be able to create a deposit even when execution of any deposits isn't occurring.

Recommendation

Consider disallowing initiating wrappings when deposit creation or execution is disabled on GMX. Otherwise, clearly document this behavior to users and monitor when features are disabled.

Resolution

Pending resolution.

L-1 | Inefficient Execution Fee Handling

CATEGORY	SEVERITY	LOCATION	STATUS
Superfluous Code	LOW	GmxV2Library.sol: 85-88	Resolved

Description

The GMX V2 system accepts and stores the execution fee as wrapped native tokens. The `exchangeRouter.sendWnt` function simply wraps the `msg.value` sent into native tokens and transfers them to the vault. Therefore it is unnecessary to unwrap the wrapped native tokens only for the `sendWnt` function to wrap them again.

```
exchangeRouter.sendWnt
msg.value
sendWnt
```

Recommendation

Transfer the execution fee into the GMX V2 system with the `exhangeRouter.sendTokens` function.

```
exhangeRouter.sendTokens
```

Resolution

Pending resolution.

L-2 | Incorrect maxWei Limit Bypass Check

CATEGORY	SEVERITY	LOCATION	STATUS
Logical Error	LOW	UpgradeableAsyncIsolationModeWrapperTrader.sol: 415	Resolved

Description

When receiving excess GM after a deposit, if it surpasses or is equal to the maximum allowed value for a market, then it is entirely sent to the vault owner. The check incorrectly includes equality with maximum WEI, since Dolomite allows deposits up to the maximum WEI, but not exceeding it.

Recommendation

Modify the comparison in the if of the `_depositIntoDefaultPositionAndClearDeposit` function from `>= maxWei` to `> maxWei`.

```
_depositIntoDefaultPositionAndClearDeposit
>=
          maxWei
> maxWei
```

Resolution

Pending resolution.

L-3 | Redundant Self Import

CATEGORY	SEVERITY	LOCATION	STATUS
Superfluous Code	LOW	GmxV2Library.sol: 24	Resolved

Description

In the `GmxV2Library.sol` file, the library itself is reimported redundantly.

```
GmxV2Library.sol
```

Recommendation

Remove the self import from line 24.

Resolution

Pending resolution.

L-4 | GM Price Impact May Be Misrepresented

CATEGORY	SEVERITY	LOCATION	STATUS
Oracle Risk	LOW	GmxV2PriceOracle.sol: 173-180	Resolved

Description

The GM oracle calculates the price impact to appropriately adjust GM's price when the price impact is negative. However, the calculation is done using the short token as the output amount.

Recommendation

Consider adjusting the `_getAdjustedAccountValues` logic to take into account the output token during a liquidation and/or documenting this behavior to users.

```
_getAdjustedAccountValues
```

Resolution

Dolomite Team: We have decided to remove the price impact calculation and instead increase the liquidation penalty.

L-5 | Incorrect Interface Used

CATEGORY	SEVERITY	LOCATION	STATUS
Typo	LOW	GenericTraderProxyBase.sol: 354-355	Resolved

Description

When calculating the actions length, the `traderType` is `IsolationModeWrapper` but the `IIsolationModeUnwrapperTrader` interface is used.

```
traderType
IsolationModeWrapper
IIsolationModeUnwrapperTrader
```

Recommendation

Use the `IIsolationModeWrapperTrader` interface for consistency.

```
IIsolationModeWrapperTrader
```

Resolution

Pending resolution.

L-6 | Dolomite Assumes Owner Can Handle GM

CATEGORY	SEVERITY	LOCATION	STATUS
Documentation	LOW	UpgradeableAsyncIsolationModeWrapperTrader.sol: 422	Resolved

Description

In the case that the deposit of excess GM into the borrow account fails, and the market supply caps are exceeded, Dolomite transfers those GM tokens directly to the vault owner.

Recommendation

Document to users this behavior to prevent unexpected loss of funds.

Resolution

Pending resolution.

L-7 | Inaccurate File Name

CATEGORY	SEVERITY	LOCATION	STATUS
Typo	LOW	UpgradeableAsyncIsolationModeWrapperTrader.sol: 52	Resolved

Description

The `_FILE` name is set to "IsolationModeWrapperTraderV2" which is not the actual name of the file. This differs from the standard within the `UpgradeableAsyncIsolationModeUnwrapperTrader.sol` contract where `_FILE = "UpgradeableUnwrapperTraderV2"`

```
_FILE
UpgradeableAsyncIsolationModeUnwrapperTrader.sol
_FILE = "UpgradeableUnwrapperTraderV2"
```

Recommendation

Change it to `bytes32 private constant _FILE = "UpgradeableWrapperTraderV2";`

```
bytes32 private constant _FILE =
          "UpgradeableWrapperTraderV2";
```

Resolution

Pending resolution.

L-8 | Typo

CATEGORY	SEVERITY	LOCATION	STATUS
Typo	LOW	GmxV2MarketTokenPriceOracle.sol: 163	Resolved

Description

There is a typo in the comment: “there’s on cap” should be “there’s no cap”.

Recommendation

Fix the typo as described above.

Resolution

Pending resolution.

L-9 | Inaccurate Comment

CATEGORY	SEVERITY	LOCATION	STATUS
Documentation	LOW	GmxV2MarketTokenPriceOracle.sol: 45	Resolved

Description

In the price oracle the documentation states: "/// @dev All of the GM tokens listed have, at-worst, 20 bp for the price deviation".

Recommendation

Update the comment to reflect the 25bp price deviation.

Resolution

Pending resolution.

Summary of Recommendations

Based on our comprehensive audit, we provide the following prioritized recommendations to improve the security posture of Dolomite.

Priority Matrix

ISSUE ID	TITLE	SEVERITY	PRIORITY
C-1	Liquidations Prevented With Pending Action	CRITICAL	Immediate
H-0	Withdrawals Apply _minOutputAmount To One Side	HIGH	High
H-2	Severely Undercollateralized Positions Cannot Be Liquidated	HIGH	High
H-3	Withdrawal Keys Misused by Differing Subaccount in Liquidations	HIGH	High
M-0	Vault Owner Can Cancel Liquidation	MEDIUM	Medium
M-1	Withdrawal Not Necessarily 50-50	MEDIUM	Medium
M-2	Excess GM Not Partially Deposited On Supply Cap	MEDIUM	Medium
M-3	Can't Unfreeze Vault If Execution Interrupted	MEDIUM	Medium
M-4	Lack Of Liquidation Incentives	MEDIUM	Medium
M-5	Withdrawals Fail When A Backing Token is Zero	MEDIUM	Medium

General Security Best Practices

- ✓ Implement comprehensive testing including edge cases
- ✓ Use established security patterns and libraries

Audit Team

Team Credentials

Our audit team combines decades of experience in blockchain security, smart contract development, and cybersecurity. Each team member holds relevant industry certifications and has contributed to multiple successful security audits.

Methodology & Standards

Our audit methodology follows industry best practices and standards:

- ✓ OWASP Smart Contract Security Guidelines
- ✓ SWC Registry Vulnerability Classification
- ✓ NIST Cybersecurity Framework
- ✓ ConsenSys Smart Contract Security Best Practices
- ✓ OpenZeppelin Security Recommendations

Audit Process

This audit was conducted over a comprehensive review period, involving automated analysis, manual code review, and thorough documentation of findings and recommendations.

Disclaimer & Legal Notice

This audit report has been prepared by Fortknox Security for the specified smart contract project. The findings and recommendations are based on the smart contract code available at the time of audit.

Scope Limitations

- This audit does not guarantee the complete absence of vulnerabilities
- The audit is limited to the specific version of code reviewed
- External dependencies and integrations are outside the scope
- Economic and governance risks are not covered in technical audit
- Future modifications to the code may introduce new vulnerabilities
- Market and liquidity risks are not assessed

Liability Statement

Fortknox Security provides this audit report for informational purposes only. We do not provide any warranties, express or implied, regarding:

- The absolute security of the smart contract
- The economic viability of the project
- The legal compliance in any jurisdiction
- Future performance or behavior of the contract
- Third-party integrations or dependencies

Legal Terms & Usage Rights

Usage Rights

This audit report may be used by the client for:

- ✓ Public disclosure and transparency
- ✓ Marketing and promotional materials
- ✓ Investor due diligence processes
- ✓ Regulatory compliance documentation
- ✓ Technical documentation and reference
- ✓ Security assessment presentations
- ✓ Community transparency initiatives

Restrictions

The following restrictions apply to this report:

- ✓ Report content may not be modified or altered
- ✓ Fortknox Security branding must remain intact
- ✓ Partial excerpts must maintain context and accuracy
- ✓ Commercial redistribution requires written permission
- ✓ Translation must preserve technical accuracy

Intellectual Property

This report contains proprietary methodologies and analysis techniques developed by Fortknox Security. The format, structure, and analytical approach are protected intellectual property.

Contact Information

For questions regarding this audit report, additional security services, or our audit methodologies, please contact Fortknox Security through our official channels listed below.

Fortknox Security

🌐 https://www.fortknox-security.xyz

🐦 @FortKnox_sec

📧 support@fortknox-security.xyz

FORTKNOX SECURITY

Web3 Security at Fort Knox Level

Contact Us

🐦 @FortKnox_sec

📱 @FortKnox_sec

🌐 fortknox-security.xyz

📧 support@fortknox-security.xyz

Audit performed by
Fortknox Security