HackSafe

Smart Contract Security Audit Report

GMX

August 2022

Security Status

★ ★ ★ ⯪ ☆

www.hacksafe.io

Audit Details

Audited project
GMX

Deployer address
0x62edc0692BD897D2295872a9FFCac5425011c661

Client contacts
GMX Token team

Blockchain
Avalanche

Website
https://gmx.io/

Disclaimer

This is a limited report on our findings based on our analysis, in accordance with good industry practice as at the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, the details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report. While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the below disclaimer below – please make sure to read it in full.

DISCLAIMER: By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice. No one shall have any right to rely on the report or its contents, and TechRate and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (HackSafe) owe no duty of care towards you or any other person, nor does HackSafe make any warranty or representation to any person on the accuracy or completeness of the report. The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and HackSafe hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, HackSafe hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against HackSafe, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report.

The analysis of the security is purely based on the smart contracts alone. No applications or operations were reviewed for security. No product code has been reviewed.

Procedure

Step 1 - In-Depth Manual Review

Manual line-by-line code reviews to ensure the logic behind each function is sound and safe from various attack vectors. This is the most important and lengthy portion of the audit process (as automated tools often cannot find the nuances that lead to exploits such as flash loan attacks).

Step 2 - Automated Testing

Simulation of a variety of interactions with your Smart Contract on a test blockchain leveraging a combination of automated test tools and manual testing to determine if any security vulnerabilities exist.

Step 3 – Leadership Review

The engineers assigned to the audit will schedule meetings with our leadership team to review the contracts, any comments or findings, and ask questions to further apply adversarial thinking to discuss less common attack vectors.

Step 4 - Resolution of Issues

Consulting with the team to provide our recommendations to ensure the code's security and optimize its gas efficiency, if possible. We assist project team's in resolving any outstanding issues or implementing our recommendations.

Step 5 - Published Audit Report

Boiling down results and findings into an easy-to-read report tailored to the project. Our audit reports highlight resolved issues and any risks that exist to the project or its users, along with any remaining suggested remediation measures. Diagrams are included at the end of each report to help users understand the interactions which occur within the project.

Background

HackSafe was commissioned by GMX to perform an audit of smart contract:

- https://snowtrace.io/address/0x62edc0692bd897d2295872a9ffcac5425011c661#code

The purpose of the audit was to achieve the

- Ensutre that the smart contract functions as intended.
- Identify potential security issues with the smart contract.

The information in this report should be understand the risk exposure of the smart contract, and as a guide to improve the security posture of the smart contract by remediating the issues that were identified.

Contract Details

Token contract details for 09.08.2022

Token Type	: ERC20
Contract name	: GMX
Contract address	: 0x62edc0692BD897D2295872a9FFCac5425011c661
Compiler version	: v0.6.12+commit.27d51765
Total supply	: 699,489.344712
Token Ticker	: GMX
Decimals	: 18
Token Holders	: 1,546
Top 100 token holder's dominance	: 99.25%
Transactions count	: 172,021
Contract deployer address	: 0x62edc0692BD897D2295872a9FFCac5425011c661
Gov owner	: 0x0339740d92fb8baf73bab0e9eb9494bc0df1cafd

Social profiles

Twitter profile	: https://twitter.com/GMX_IO
Github profile	: https://github.com/gmx-io
Telegram profile	: https://t.me/GMX_IO

Audit Summary

According to the standard audit assessment, Customer`s solidity smart contracts are “poor”. This token contract does contain owner control, which do not make it fully decentralized as owner does have control over smart contract.

Insecure	Poor secured	Secure	Well-secured

You are here 

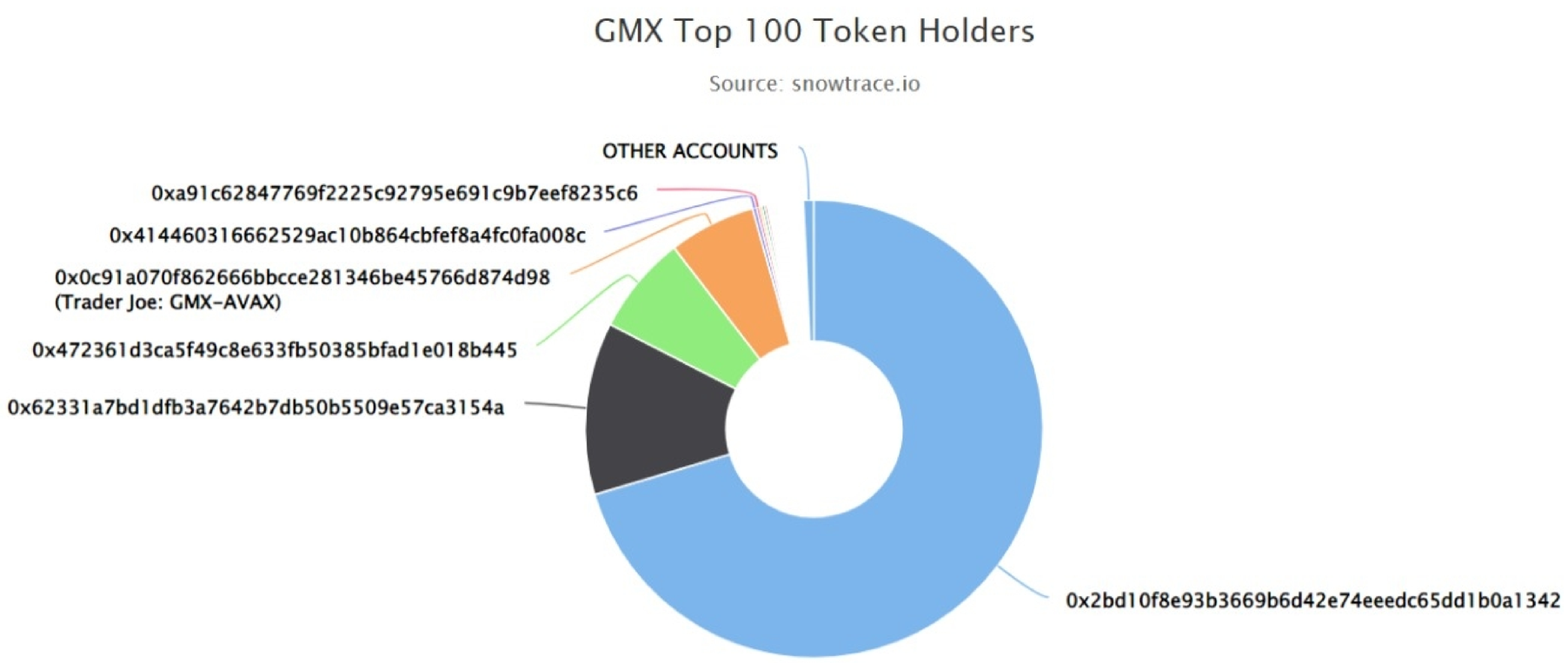
We used various tools like Slither, Mythril and Remix IDE. At the same time this finding is based on critical analysis of the manual audit. All issues found during automated analysis were manually reviewed and applicable vulnerabilities are presented in the issues checking status.

We found 1 critical, 0 high, 0 medium and 0 low and some very low-level issues. These issues are not critical ones.

GMX Token Distribution

The top 100 holders collectively own 99.26% (695,057.75 Tokens) of GMX

Token Total Supply: 700,260.81 Token | Total Token Holders: 1,542



GMX Token Top 20 Token Holders

(A total of 695,057.75 tokens held by the top 100 accounts from the total supply of 700,260.81 token)

Rank	Address	Quantity (Token)	Percentage
1	0x2bd10f8e93b3669b6d42e74eeedc65dd1b0a1342	492,806.265696672699699641	70.3747%
2	0x62331a7bd1dfb3a7642b7db50b5509e57ca3154a	85,299.103285640780514263	12.1810%
3	0x472361d3ca5f49c8e633fb50385bfad1e018b445	48,871.153009759331650753	6.9790%
4	Trader Joe: GMX-AVAX	43,040.380910310692933637	6.1463%
5	0x414460316662529ac10b864cbfef8a4fc0fa008c	1,863.709414597177546435	0.2661%
6	0xa91c62847769f2225c92795e691c9b7eef8235c6	1,587.564320145166779137	0.2267%
7	0xa5c41107c34041a658ea831bdea91e55a067b7ea	1,392.733394795062110046	0.1989%
8	0xf6a6ddeb69fcda0a93e3269dfb03876944db1152	1,291.470824772427222146	0.1844%
9	0x34cae1d9e2d014b7b9e6295c66c554d7e79713d3	1,219.806502453853159734	0.1742%
10	0xc73604f8155d33791d3767e39dca7eebfdfba9ba	1,016.705563875718119231	0.1452%
11	Synapse: Bridge	994.674241557236368581	0.1420%
12	0xc27e87cfe1fd2ed6f43dfffbb9e9e46428497a24	869.626182304920766105	0.1242%
13	0xb428e90854d9539b0a6f2a05cf334ce303fc9618	806.387983275037719376	0.1152%
14	0x861c27b7c88aa468b757c58cbbf186e92ebfade9	777.640961231767183969	0.1111%
15	0x19dde5f247155293fb8c905d4a400021c12fb6f0	751.855184862904107743	0.1074%
16	0x47d766b2d229d0323f36660fa9447bedaf6d9946	709.282767359809843417	0.1013%
17	0x69db2c89974f74b4a16e54eb75884465a55f3980	700.523498448456110387	0.1000%
18	0x4519674573d0d04268468430fb4b3a8feab894e8	687.196659859316775436	0.0981%
19	0x25b3d855672cfddaeef0f8956a9d4ca58e8061c5	533.71	0.0762%
20	0x527788ae179be743614496680d38b39d87ee1ce8	416.631288794898419864	0.0595%

Contract functions details

GMX.sol
+ GMX (MintableBaseToken)
    -[Pub] <constructor>
    -[Ext] id

MintableBaseToken.sol
+ MintableBaseToken (BaseToken, IMintable)
    -[Pub] <constructor>
    -[Ext] setMinter
      -modifiers: onlyGov
    -[Ext] mint
      -modifiers: onlyMinter
    -[Ext] burn
      -modifiers: onlyMinter

BaseToken.sol
+ BaseToken (IERC20, IBaseToken)
    -[Pub] <constructor>
    -[Ext] setGov #
      -modifiers: onlyGov
    -[Ext] setInfo #
      -modifiers: onlyGov
    -[Ext] setYieldTrackers #
      -modifiers: onlyGov
    -[Ext] addAdmin #
      -modifier: onlyGov
    -[Ext] removeAdmin #
      -modifier: onlyGov
    -[Ext] withdrawToken #
      -modifier: onlyGov
    -[Ext] setInPrivateTransferMode #
      -modifier: onlyGov
    -[Ext] setHandler #
      -modifier: onlyGov
    -[Ext] addNonStakingAccount #
      -modifier: onlyAdmin
    -[Ext] removeNonStakingAccount #
      -modifier: onlyAdmin
    -[Ext] recoverClaim #
      -modifier: onlyAdmin

Contract functions details

-[Ext] claim #
-[Ext] totalStaked
-[Ext] balanceOf
-[Ext] stakedBalance
-[Ext] transfer #
-[Ext] allowance
-[Ext] approve #
-[Ext] transferFrom #
-[Int] _mint #
-[Int] _burn #
-[Pvt] _transfer #
-[Pvt] _approve #
-[Pvt] _updateRewards #

IMintable.sol
+[Int] IMintable
-[Ext] isMinter
-[Ext] setMinter
-[Ext] mint
-[Ext] burn

SafeMath.sol
+[Lib] SafeMath
-[Int] add
-[Int] sub
-[Int] sub
-[Int] mul
-[Int] div
-[Int] div
-[Int] mod

IERC20.sol
+ [Int] IERC20
-[Ext] totalSupply
-[Ext] balanceOf
-[Ext] transfer
-[Ext] allowance
-[Ext] approve
-[Ext] transferFrom

Contract functions details

SafeERC20.sol
+[Lib] SafeERC20
    -[Int] safeTransfer
    -[Int] safeTransferFrom
    -[Int] safeApprove
    -[Int] safeIncreaseAllowance
    -[Int] safeDecreaseAllowance
    -[Pvt] _callOptionalReturn

IYieldTracker.sol
+[Int] IYieldTracker
    -[Ext] claim
    -[Ext] updateRewards
    -[Ext] getTokensPerInterval
    -[Ext] claimable

IBaseToken.sol
+[Int] IBaseToken
    -[Ext] totalStaked
    -[Ext] stakedBalance
    -[Ext] removeAdmin
    -[Ext] setInPrivateTransferMode
    -[Ext] withdrawToken

Address.sol
+ [Lib] Address
    -[Int] isContract
    -[Int] sendValue
    -[Int] functionCall
    -[Int] functionCallWithValue
    -[Int] functionCallWithValue
    -[Int] functionStaticCall
    -[Int] functionStaticCall
    -[Int] functionDelegateCall
    -[Int] functionDelegateCall
    -[Pvt] _verifyCallResult

($) = payable function
= non-constant function

Issues Checking Status

No.	Title	Status
1.	Unlocked Compiler Version	Passed
2.	Missing Input Validation	Passed
3.	Race conditions and Reentrancy. Cross-function race conditions.	Passed
4.	Possible delays in data delivery	Passed
5.	Oracle calls.	Passed
6.	Timestamp dependence.	Passed
7.	Integer Overflow and Underflow	Passed
8.	DoS with Revert.	Passed
9.	DoS with block gas limit.	Passed
10.	Methods execution permissions.	Critical issue
11.	Economy model of the contract.	Passed
12.	Private use data leaks.	Passed
13.	Malicious Event log.	Passed
14.	Scoping and Declarations.	Passed
15.	Uninitialized storage pointers.	Passed
16.	Arithmetic accuracy.	Passed
17.	Design Logic.	Passed
18.	Safe Open Zeppelin contracts implementation and usage.	Passed
19.	Incorrect Naming State Variable	Passed
20.	Too old version	Passed

Severity Definitions

Risk Level	Description
Critical	Critical vulnerabilities are usually straightforward to exploit and can lead to assets loss or data manipulations.
High	High-level vulnerabilities are difficult to exploit; however, they also have a significant impact on smart contract execution, e.g., public access to crucial functions
Medium	Medium-level vulnerabilities are important to fix; however, they can't lead to assets loss or data manipulations.
Low	Low-level vulnerabilities are mostly related to outdated, unused, etc. code snippets that can't have a significant impact on execution.

Security Issues

Critical Severity Issues

One critical severity issue found.

1. Method execution permission

Unused function.

- **Description**

  Function burn in MintableBaseToken.sol contract have modifier named onlyMinter means it is only called by only minter of the contract. Function can burn tokens of any of the user.

- **Recommendation**

  It is highly advisable to remove _account parameter of the function and set it to msg.sender.

High Severity Issues

No high severity issue found.

Medium Severity Issues

No medium severity issues found.

Low Severity Issues

No low severity issue found.

Centralization

Owner privileges:

- GMX Contract:
    - Gov. privileges
    - Gov can set minter.
    - Gov can set other gov.
    - Gov can change name, symbol of token.
    - Gov can setYieldTrackers.
    - Gov can add and remove admin.
    - Gov can withdraw tokens.

This smart contract has some functions which can be executed by the Gov (Owner) only. If the admin wallet private key would be compromised, then it would create trouble but smart contract ownership has been renounced. Following are Gov functions functions:

- Addadmin
- Setgov
- Setinfo
- Setyieldtrackers
- Addadmin
- Removeadmin
- Withdrawtoken
- Setinprivatetransfer
- mode
- Sethandler

Conclusion

Smart contract contains Critical severity issues! The further transfer and operations with the fund raised are not related to this particular contract.

HackSafe note: Please check the disclaimer above and note, the audit makes no statements or warranties on business model, investment attractiveness or code sustainability. The report is provided for the only contract mentioned in the report and does not include any other potential contracts deployed by Owner.