DcentraLab Diligence

dcentralab.com/diligence

Poolz

Audit Report

Poolz Finance

https://www.poolz.finance

Provided By DcentraLab Diligence on March 23, 2023

Security Audit Score

Pass

DcentraLab Diligence team has conducted an extensive audit on the Poolz Token code and has found it to represent adequate code quality and a low risk level given proper deployment and multi-sig permissioning.

- Low Risk
- Small Risk
- Medium Risk
- High Risk

Audit Report | POOLX Token

CodeBase:

https://github.com/The-Poolz/PoolzToken

Initial Commit Hash:

9b012e9042b9f65165895847859bf4c6116257f1

Fixes Commit Hash:

3c0f2e003b96ed54f4aa99747ac22d3c27a6ed08

Final Report Git Tag: V1.1.3 (Diff View)

https://github.com/The-Poolz/PoolzToken/compare/V1.1.0...V1.1.3

Contracts Audited:

ERC20BurnableMintableCapped.sol
ERC20Mintable.sol
Roles.sol
POOLZ.sol

General Recommendations:

Increase pragma version to ^0.8.4 in order to take advantage of the newest gas/bytecode optimizations.

Risks:

DcentraLab Diligence (DD) has performed all checks and verifications in its capacity to ascertain the safety of the code. However, it should be noted that misuse of the code, bad deployment practices, bad key management, exposing of private keys of the deployer and/or owner address and/or multi-sig signer addresses and/or fee collector address and/or any exposition of the code to malicious actors may result in an exploit of the code and loss of state and/or funds. Furthermore, there is always a chance that other Smart Contracts code could be written and deployed to cause the provided code by DD to act outside the intended scope by the client, to the point of causing state corruption or loss of funds to the client of the users of the code.

Issues Severity Reference Table

Type

Informational

This issue is not critical and does not pose an immediate threat to the functionality or security of the smart contract. It is simply an informational item that the auditors have identified and recommends addressing for best practices or to improve the overall performance of the contract.

Low

This issue is relatively minor and does not pose a significant risk to the functionality or security of the smart contract. While it is recommended to address these issues to ensure the highest level of quality and security, they are not likely to cause significant problems if left unaddressed.

Medium

This issue poses a moderate risk to the functionality or security of the smart contract. While it may not be immediately exploitable, it has the potential to cause problems in the future if left unaddressed. It is recommended to address these issues as soon as possible to prevent any potential negative impact on the contract.

High

This issue poses a significant risk to the functionality or security of the smart contract. Addressing these issues as soon as possible is recommended to prevent any potential negative impact on the contract. Failure to address these issues could result in significant problems and potential loss of funds or other assets.

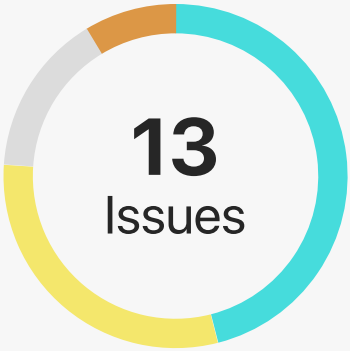
Critical

This issue poses an immediate and severe risk to the functionality or security of the smart contract. It is recommended to address these issues immediately to prevent any potential negative impact on the contract. Failure to address these issues could result in catastrophic problems and significant loss of funds or other assets.

Discussion

The issue severity is dependent on design, centralization, and product specifications of the project.

Findings Summary



Informational
High Risk
Medium Risk
Discussion

ID	Title	Severity	Status
1	Role struct	Informational	Resolved
2	Roles library internal usage	Informational	Partially Resolved
3	_addMinter and _removeMinter functions visibility	Informational	Resolved
4	Impossible renouncement of malicious minter	Medium	Partially Resolved
5	Mint function unnecessary virtualization	Informational	Resolved
6	Potential irreversible lock on minting	Medium	Partially Resolved
7	Missing deployment script verifying proper access control	High	Partially Resolved
8	Missing Implementation of Multi-Sig	Medium	Partially Resolved
9	Missing Upgradability Schema	Discussion	Acknowledged

Findings Summary

ID	Title	Severity	Status
10	Missing Security Measures	Discussion	Acknowledged
11	Potentially volatile use of Roles	Medium	Partially Resolved
12	Unconventional naming	Informational	Acknowledged
13	Event won't get emitted on initial minter addition	Informational	Acknowledged

Complete Analysis

ID 1: **Status: Resolved**

Informational | Role struct

Roles library:
Description: While defining a struct that contains only one mapping inside a library is logical due to the inability of libraries to contain descriptions of simple types, it is unnecessarily increasing code complexity and a number of performed operations.

Recommendation: We recommend parsing bearers mapping directly as a functions argument and removing Role struct as this will reduce the number of operations needed to access the mapping values.

ID 2: **Status: Partially Resolved**

Informational | Roles library internal usage

Roles library:
Description: Roles library consists solely of internal functions, which means that it will be an integrated part of token bytecode. We consider this as an unnecessary bytecode increase.

Recommendation: Prepare library for separate deployment in order to optimize contract code by changing function visibility to external. This way library bytecode will not be directly contained inside token logic.

Fixes Feedback: The visibility of the Roles functions has indeed been changed to external. However, Roles is now a contract that is inherited by ERC20Mintable, so it is still an integrated part of the token bytecode.

Complete Analysis

ID 3: **Status: Resolved**

Informational | _addMinter and _removeMinter functions visibility

ERC20Mintable contract:
Description: Mentioned functions act as a middleware between their accessors which is not necessary in this case.

Recommendation: Events from functions can be moved to the library and used in library function logic. This way, functions will still be directly accessible to the constructor, and external functions defined in ERC20BurnableMintableCapped.sol, and internal functions in ERC20Mintable.sol can be removed.

ID 4: **Status: Partially Resolved**

Medium | Impossible renouncement of malicious minter

ERC20BurnableMintableCapped contract:
Description: In case of minter wallet being taken over there is no possibility to remove its role, as minter can only renounce himself.

Recommendation (+): In order to make your role system safer, take a look at AccessControl.sol by openzeppelin. This contract is considered a standard for role-based security systems in EVM dApps. In that system, there's an admin role that can, as a higher authority, apply and renounce roles of other accounts in the system. One of the best solutions could be to create an admin role and assign it to a multi-signature wallet owned by your organization, or just use the AccessControl.sol solution directly as that is considered safest and most optimized gas wise. In case you choose the second, library issues found are not applied anymore.

Complete Analysis

Fixes Feedback: No code changes were made.

Comment: Team did set multi-sig contract as an only minter, which is their approach to solving this problem. While this solution makes contract more secure in current conditions, we are marking this issue as "Partially Resolved" as there is no safety improvement to the code.

ID 5: **Status: Resolved**

Informational | Mint function unnecessary virtualization

ERC20BurnableMintableCapped contract:
Description: Mint function is not overridden in POOLZ.sol contract, which inherits the contract to which this paragraph is dedicated.

Recommendation: Remove virtual keyword from mint function if you are not planning to override it in the future (this relates to some other contract that would inherit this logic, as current architecture is non-upgradeable).

We recommend you reading the following:

- From pragma 0.8.4 you are able to use custom errors which optimize gas usage and bytecode size: https://blog.soliditylang.org/2021/04/21/custom-errors/
- This really informative article about libraries: https://jeancvllr.medium.com/solidity-tutorial-all-about-libraries-762e5a3692f9
-

Fixes Feedback: Code was changed in such a way that this issue is no longer applicable.

Complete Analysis

ID 6: **Status: Partially Resolved**

Medium | Potential irreversible lock on minting

ERC20BurnableMintableCapped contract:
Description: renounceMinter() - It is possible to have no minters by mistake, and this is irreversible. If there is only one minter, and they want to add another minter, they may mistakenly call the renounceMinter() function first, resulting in no one being able to add another minter.

Recommendation: add an address parameter for the renounceMinter() function. Inside the function implementation, add the new minter by calling _addMinter(). If you wish to just remove a minter, simply call the function with a zero address parameter.
Notice: do not call _addMinter() if the parameter is zero address.

Fixes Feedback: No code changes were made.

Comment: While this flow of events is still possible, the project considers this issue inapplicable to their use cases. With using only one minter, which is a multi-signature wallet, the issue gets marked as “Partially Resolved” as there is no in-code prevention of described undesired behavior.

ID 7: **Status: Partially Resolved**

High | Missing deployment script verifying proper access control

Description: The repo is missing a deployment script that will effectively remove the hot wallet as owner and minter. Without this, it is not possible to ascertain that minting will not be compromised.

Complete Analysis

Recommendation: Please add a full deployment script that removes deployer (which is usually hot address) as owner and minter, and applies a multi-sig / properly guarded address as minter.

Comment: Project team decided to continue forward without providing a deployment script. Deployment was done successfully so the issue remains marked as "Partially resolved."

ID 8: **Status: Partially Resolved**

Medium | Missing Implementation of Multi-Sig

Description: If minting is intended to be permissioned by a multi-sig, as the code for such is not in the scope of the audit, it's not possible to verify that ownership and minting permissions will be well kept, potentially exposing the token to exploit.

Recommendation: Include the multi-sig code in the audit, or utilize a well-established, battle-tested implementation of a multi-sig (e.g., Gnosis)

Comment: Project decided not to add the multi-sig contract to the audit scope, but attests that the Token has been deployed with a multi-sig contract as sole minter, and that the hot deployer role as minter has been removed:

Deployed Token on Ethereum: https://etherscan.io/token/0xeef66125bfcffdb1642c7e85a432cd1b78038399

Multisig Contract: https://github.com/The-Poolz/MultiSighttps://etherscan.io/address/0x3e4588c3C4E6ff3da84ab5401490d9c9eA820d3E

Adding Multisig as Minter: https://etherscan.io/tx/0xcdf563f5dce5e98f90c2cf0eb96725a3abe5ac6a794976416c65157f08559c7f

Remove Hot Wallet as Minter: https://etherscan.io/tx/0x9ac9e5e8c14ac4d89a0494ced7e2ce794a6fe053c2931f0c20488eb8a36d7c15

Complete Analysis

ID 9: **Status: Acknowledged**

Discussion | **Missing Upgradability Schema**

Description: Current contract is not upgradeable, limiting team ability to respond in case of requirement for adding new features (e.g. taxation, security measures). On the other hand, token being non-upgradable is seen at times to be more secure.

Recommendation: Enable upgradability with high level security multi-sig vault or DAO contract of token holders as owner of proxy admin for upgradability, to keep both security and flexibility and adaptability to changing conditions in market, token use case, client product, tokenomics etc.

Comment: Team is intentionally using the non-upgradeable architectural scheme.

ID 10: **Status: Acknowledged**

Discussion (Low to Medium) | **Missing Security Measures**

Description: Current token does not hold security measures capabilities, e.g., freeze token, blacklist address, destroy black funds, etc. Some might see such capabilities as centralized and unwanted, but they are very useful to mitigate emergencies, and can be set to be controllable by a highly secure multi-sig.

Recommendation: Discuss internally and decide whether to add such features, or enable upgradability to allow potential to add such later. In general, we recommend tokens to have such capabilities owned by a highly secure multi-sig.

Comment: This way of functioning was intentionally implemented by the team. They're accepting this architectural risk

Complete Analysis

ID 11: **Status: Unresolved**

Medium | Potentially volatile use of Roles

Roles library:

Description: Currently, a lib for roles is inefficiently being used and only allows in indirect manner to specify a list of allowed minters used in another contract. Beyond potential state manipulation risks and inefficiencies of using complex mappings to store a simple list/mapping of minters, having a complex setup leads to more exposure also on security side.

Recommendations:

- Manage minter privileges directly on the EC20Mintable contract.
- Enable only multi-sig owner to mint , and use simple address state to track the single minter (owner), which can upgrade itself only
- If absolutely required to appoint more than one minter or non multi-sig minter, allow only multi-sig owner to appoint minters, don't allow minters to appoint other minters arbitrarily.

Fixes Feedback: Sub-issues of number 2 and 3 were not fixed. There is no proper privilege segregation implemented and therefore issues connected with minter role management still persist.

New Issues Post Audit:

ID 12: **Status: Acknowledged**

Informational | Unconventional naming

Description: Unconventional variable naming present at line 9 @ MinterRole.sol

Recommendation: Apply valid naming convention to the variable.

ID 13: **Status: Acknowledged**

Informational | Event won't get emitted on initial minter addition

Description: During initial minter addition in a constructor, an event will not emit on state change on line 10 @ ERC20Mintable.sol

Recommendation: Implement logic in such a way that event reflects the state change.

Disclaimer:

DcentraLab Diligence (DD) has provided the code to the client as is and assumes no responsibility nor legal liability for any use client may do with the code. Any and all usage and/or deployment of the code provided by DcentraLab Diligence will be done solely by the client, at the sole discretion, responsibility, risk, and legal liability of the Client, and DD will not be held accountable or liable for any loss of funds, security exploits or incidents, or any other unintended or negative outcome that may occur in relation to the code provided by DD.

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 DD 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 and the provided code or services as part of the SOW pertaining to 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 it 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. DD's position is that each company and individual are responsible for their own due diligence and continuous security. DD'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 DD are subject to dependencies and are under continuing development. You agree that your access and/or use, including but not limited to any services, code, 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, DcentraLab Diligence (DD) HEREBY DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE SERVICES, ASSESSMENT REPORT, OR OTHER MATERIALS. WITHOUT LIMITING THE FOREGOING, DD SPECIFICALLY DISCLAIMS

ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT, AND ALL WARRANTIES ARISING FROM THE COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

WITHOUT LIMITING THE FOREGOING, DD MAKES NO WARRANTY OF ANY KIND THAT THE SERVICES, THE LABELS, THE ASSESSMENT / VERIFICATION REPORT, WORK PRODUCT, CODE OR OTHER MATERIALS, OR ANY PRODUCTS OR RESULTS OF THE USE THEREOF, WILL MEET THE 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 DISCLAIMER [ASSESSMENT NAME] FOREGOING, DD PROVIDES NO WARRANTY OR UNDERTAKING, AND MAKES NO REPRESENTATION OF ANY KIND THAT THE SERVICE WILL MEET THE 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 DD NOR ANY OF DD'S AGENTS MAKES ANY REPRESENTATION OR WARRANTY OF ANY KIND, EXPRESS OR IMPLIED AS TO THE ACCURACY, RELIABILITY, OR CURRENCY OF ANY INFORMATION, CODE OR CONTENT PROVIDED THROUGH THE SERVICE. DD 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 CODE, OR (II) ANY PERSONAL INJURY OR PROPERTY DAMAGE, OF ANY NATURE WHATSOEVER, RESULTING FROM CUSTOMER'S ACCESS TO OR USE OF THE SERVICES, CODE, 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 THE 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 THE 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 DD'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 DD WITH RESPECT TO SUCH SERVICES, ASSESSMENT REPORT, AND ANY ACCOMPANYING MATERIALS. THE REPRESENTATIONS AND WARRANTIES OF DD CONTAINED IN THIS AGREEMENT ARE SOLELY FOR THE BENEFIT OF THE 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 DD 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, CODE, OR MATERIALS, SHALL NOT BE CONSIDERED OR RELIED UPON AS ANY FORM OF FINANCIAL, TAX, LEGAL,
REGULATORY, OR OTHER ADVICE.

dcentralab.com/diligence

DcentraLab Diligence

Provided By DcentraLab Diligence on March 23, 2023