Calgo(CLGO)
Smartcontract Security
Audit Report

2024. 09

From CloudGuard Corp.,

2024. 09

Document History

Date	Name	History
2024.09.09	CloudGuard	Initial

Project outline

1.1. Purpose

The purpose of this inspection is to conduct a security audit on the [CLGO] Smartcontract to discover potential hacking weaknesses, analyze the cause, and respond

1.2. Target

The subjects of this inspection are as follows.

No	Category	Addr	Memo
1	Smartcontract	0x6F7E8CE7Db573613f4422B71dA2071869A48c5De	ETH Mainnet

1.3. Schedule

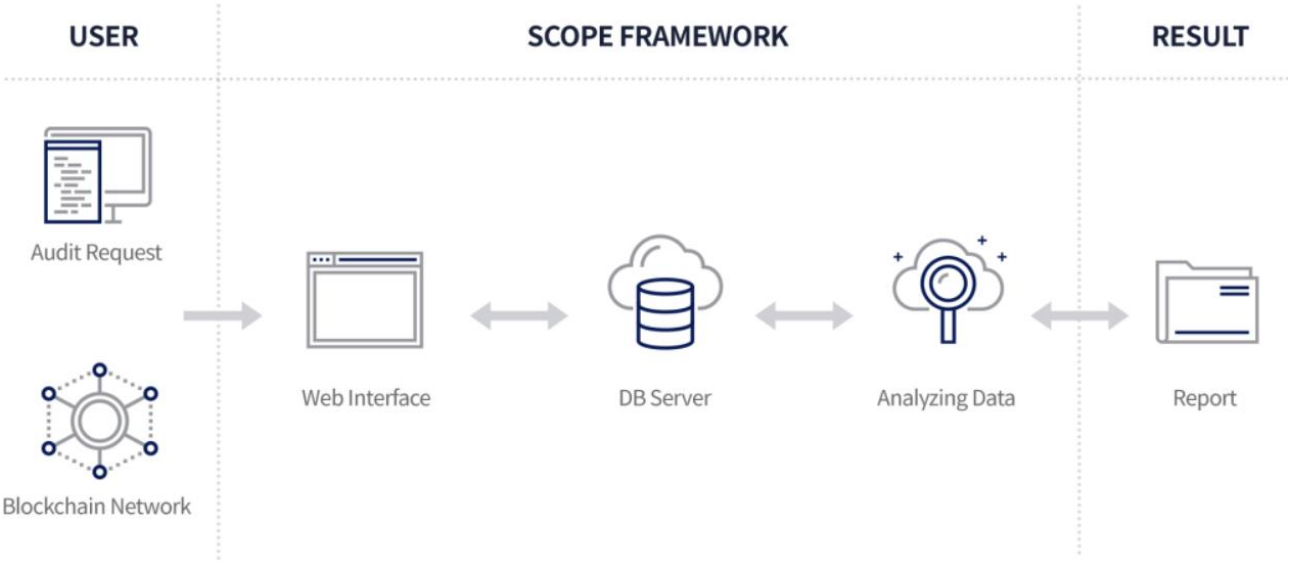
Work	Detail	Timeline	Memo
business consultation	Build Environment	1 day	
Audit	Smartcontract static auditing	2 days	
	Smartcontract Dynamic Auditing	3 days	
Report / review	Report	1 day	
	Review	1 day	

1.4. Environment

Work	Name	Platform	Memo
Audit	Smartcontract Audit	-	

Process

1.5. Process Detail



1.6. Check List

No	Detector	What Detects	Impact	external-function
1	`abiencoderv2-array`	Storage abiencoderv2 array	High	High
2	`arbitrary-send-erc20`	transferFrom uses arbitrary from	High	High
3	`array-by-reference`	Modifying storage array by value	High	High
4	`encode-packed-collision`	ABI encodePacked Collision	High	High
5	`incorrect-shift`	The order of parameters in a shift instruction is incorrect.	High	High
6	`multiple-constructors`	Multiple constructor schemes	High	High
7	`name-reused`	Contract's name reused	High	High

8	protected-vars	Detected unprotected variables	High	High
9	public-mappings-nested	Public mappings with nested variables	High	High
10	rtlo	Right-To-Left-Override control character is used	High	High
11	shadowing-state	State variables shadowing	High	High
12	suicidal	Functions allowing anyone to destruct the contract	High	High
13	uninitialized-state	Uninitialized state variables	High	High
14	uninitialized-storage	Uninitialized storage variables	High	High
15	unprotected-upgrade	Unprotected upgradeable contract	High	High
16	codex	Use Codex to find vulnerabilities.	High	Low
17	arbitrary-send-erc20-permit	transferFrom uses arbitrary from with permit	High	Medium
18	arbitrary-send-eth	Functions that send Ether to arbitrary destinations	High	Medium
19	controlled-array-length	Tainted array length assignment	High	Medium
20	controlled-delegatecall	Controlled delegatecall destination	High	Medium
21	delegatecall-loop	Payable functions using delegatecall inside a loop	High	Medium
22	incorrect-exp	Incorrect exponentiation	High	Medium
23	incorrect-return	If a return is incorrectly used in assembly mode.	High	Medium
24	msg-value-loop	msg.value inside a loop	High	Medium
25	reentrancy-eth	Reentrancy vulnerabilities (theft of ethers)	High	Medium
26	return-leave	If a return is used instead of a leave.	High	Medium

27	storage-array	Signed storage integer array compiler bug	High	Medium
28	unchecked-transfer	Unchecked tokens transfer	High	Medium
29	weak-prng	Weak PRNG	High	Medium
30	domain-separator-collision	Detects ERC20 tokens that have a function whose signature collides with EIP-2612's DOMAIN_SEPARATOR()	Medium	High
31	enum-conversion	Detect dangerous enum conversion	Medium	High
32	erc20-interface	Incorrect ERC20 interfaces	Medium	High
33	erc721-interface	Incorrect ERC721 interfaces	Medium	High
34	incorrect-equality	Dangerous strict equalities	Medium	High
35	locked-ether	Contracts that lock ether	Medium	High
36	mapping-deletion	Deletion on mapping containing a structure	Medium	High
37	shadowing-abstract	State variables shadowing from abstract contracts	Medium	High
38	tautological-compare	Comparing a variable to itself always returns true or false, depending on comparison	Medium	High
39	tautology	Tautology or contradiction	Medium	High
40	write-after-write	Unused write	Medium	High
41	boolean-cst	Misuse of Boolean constant	Medium	Medium
42	constant-function-asm	Constant functions using assembly code	Medium	Medium
43	constant-function-state	Constant functions changing the state	Medium	Medium

44	divide-before-multiply	Imprecise arithmetic operations order	Medium	Medium
45	out-of-order-retryable	Out-of-order retryable transactions	Medium	Medium
46	reentrancy-no-eth	Reentrancy vulnerabilities (no theft of ethers)	Medium	Medium
47	reused-constructor	Reused base constructor	Medium	Medium
48	tx-origin	Dangerous usage of tx.origin	Medium	Medium
49	unchecked-lowlevel	Unchecked low-level calls	Medium	Medium
50	unchecked-send	Unchecked send	Medium	Medium
51	uninitialized-local	Uninitialized local variables	Medium	Medium
52	unused-return	Unused return values	Medium	Medium
53	incorrect-modifier	Modifiers that can return the default value	Low	High
54	shadowing-builtin	Built-in symbol shadowing	Low	High
55	shadowing-local	Local variables shadowing	Low	High
56	uninitialized-fptr-cst	Uninitialized function pointer calls in constructors	Low	High
57	variable-scope	Local variables used prior their declaration	Low	High
58	void-cst	Constructor called not implemented	Low	High
59	calls-loop	Multiple calls in a loop	Low	Medium
60	events-access	Missing Events Access Control	Low	Medium
61	events-maths	Missing Events Arithmetic	Low	Medium
62	incorrect-unary	Dangerous unary expressions	Low	Medium
63	missing-zero-check	Missing Zero Address Validation	Low	Medium

64	reentrancy-benign	Benign reentrancy vulnerabilities	Low	Medium
65	reentrancy-events	Reentrancy vulnerabilities leading to out-of-order Events	Low	Medium
66	return-bomb	A low level callee may consume all callers gas unexpectedly.	Low	Medium
67	timestamp	Dangerous usage of block.timestamp	Low	Medium
68	assembly	Assembly usage	Informational	High
69	assert-state-change	Assert state change	Informational	High
70	boolean-equal	Comparison to boolean constant	Informational	High
71	cyclomatic-complexity	Detects functions with high (> 11) cyclomatic complexity	Informational	High
72	deprecated-standards	Deprecated Solidity Standards	Informational	High
73	erc20-indexed	Un-indexed ERC20 event parameters	Informational	High
74	function-init-state	Function initializing state variables	Informational	High
75	incorrect-using-for	Detects using-for statement usage when no function from a given library matches a given type	Informational	High
76	low-level-calls	Low level calls	Informational	High
77	missing-inheritance	Missing inheritance	Informational	High
78	naming-convention	Conformity to Solidity naming conventions	Informational	High
79	pragma	If different pragma directives are used	Informational	High
80	redundant-statements	Redundant statements	Informational	High

81	solc-version	Incorrect Solidity version	Informational	High
82	unimplemented-functions	Unimplemented functions	Informational	High
83	unused-import	Detects unused imports	Informational	High
84	unused-state	Unused state variables	Informational	High
85	costly-loop	Costly operations in a loop	Informational	Medium
86	dead-code	Functions that are not used	Informational	Medium
87	reentrancy-unlimited-gas	Reentrancy vulnerabilities through send and transfer	Informational	Medium
88	too-many-digits	Conformance to numeric notation best practices	Informational	Medium
89	cache-array-length	Detects for loops that use length member of some storage array in their loop condition and don't modify it.	Optimization	High
90	constable-states	State variables that could be declared constant	Optimization	High
91	external-function	Public function that could be declared external	Optimization	High
92	immutable-states	State variables that could be declared immutable	Optimization	High
93	var-read-using-this	Contract reads its own variable using this	Optimization	High

1.7. Result

[Passed]

[CLGO] As a result of the Smartcontract security audit, a total of 0 vulnerabilities were found, among which 0 vulnerabilities of 'high', 0 of 'medium' vulnerabilities, 0 of 'low' vulnerabilities, and 'information' ratings were found.

NO.	Audit Items	Result
1	Replay Vulnerability	Passed
2	Denial of Service Vulnerability	Passed
3	Race Conditions Vulnerability	Passed
4	Authority Control Vulnerability	Passed
5	Integer Overflow and Underflow Vulnerability	Passed
6	Gas Optimization Audit	Passed
7	Design Logic Audit	Passed
8	Uninitialized Storage Pointers Vulnerability	Passed
9	Arithmetic Accuracy Deviation Vulnerability	Passed
10	"False top-up" Vulnerability	Passed
11	Malicious Event Log Audit	Passed
12	Scoping and Declarations Audit	Passed
13	Safety Design Audit	Passed
14	Non-privacy/Non-dark Coin Audit	Passed

Detailed results

1.8. Smartcontract

```
pragma solidity ^0.8.20;


import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Burnable.sol";
import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Pausable.sol";
import "@openzeppelin/contracts/access/AccessControl.sol";
import "@openzeppelin/contracts/token/ERC20/extensions/ERC20Permit.sol";


contract CLGO is ERC20, ERC20Burnable, AccessControl, ERC20Permit {
    bytes32 public constant MINTER_ROLE = keccak256("MINTER_ROLE");


    constructor()
        ERC20("CLGO", "CLGO")
        ERC20Permit("CLGO")
    {
        _grantRole(DEFAULT_ADMIN_ROLE, msg.sender);
        _grantRole(MINTER_ROLE, msg.sender);
    }


    function mint(address to, uint256 amount) public onlyRole(MINTER_ROLE) {
        _mint(to, amount);
    }
```