



Dust Labs

Dusties Protocol

SOLANA PROGRAM AUDIT

29.01.2023

Made in Germany by Chainsulting.de



Table of contents

1. Disclaimer.....	3
2. Project Overview	4
3. Vulnerability & Risk Level	5
4. Auditing Strategy and Techniques Applied.....	6
4.1 Methodology	6
5. Metrics	7
5.1 Tested Program Files.....	7
5.2 Used Code from other Frameworks/Dependencies.....	9
5.3 Structure.....	10
6. Scope of Work	12
6.1 Findings Overview	13
6.2 Manual and Automated Vulnerability Test.....	14
6.2.1 Integer Overflow and Underflow	14
6.2.2 Timestamp Dependencies	15
6.2.3 Compiler Error.....	17
6.2.4 Methods Execution Permissions (Access Control)	18
6.3 Verify Claims	19
7. Executive Summary.....	20
8. About the Auditor	21

1. Disclaimer

The audit makes no statements or warranties about utility of the code, safety of the code, suitability of the business model, investment advice, endorsement of the platform or its products, regulatory regime for the business model, or any other statements about fitness of the contracts to purpose, or their bug free status. The audit documentation is for discussion purposes only.

The information presented in this report is confidential and privileged. If you are reading this report, you agree to keep it confidential, not to copy, disclose or disseminate without the agreement of Dust Labs. If you are not the intended receptor of this document, remember that any disclosure, copying or dissemination of it is forbidden.

Major Versions / Date	Description
0.1 (23.01.2023)	Layout
0.4 (25.01.2023)	Automated Security Testing Manual Security Testing
0.5 (26.01.2023)	Verify Claims and Test Deployment
0.9 (27.01.2023)	Summary and Recommendation
1.0 (29.01.2023)	Final document

2. Project Overview

DUST protocol is a decentralized protocol and SPL (Solana Program Library) token created on the Solana blockchain with a starting supply of zero, and a maximum supply of 33,300,000. DUST has an emission schedule with multiple halvings and mining rewards that are earned via staking NFTs. Countless projects have independently adopted DUST within their own ecosystems making it the most used SPL token on the Solana blockchain.

A new innovation from Dust Labs is Dusties, a webapp that supports creation of auctions and raffles on the Solana blockchain.

Website: <https://www.dustprotocol.com>

Documentation: <https://docs.dustprotocol.com>

Twitter: <https://twitter.com/DeGodsNFT>

Discord: <https://discord.gg/dedao>



3. Vulnerability & Risk Level

Risk represents the probability that a certain source-threat will exploit vulnerability, and the impact of that event on the organization or system. Risk Level is computed based on CVSS version 3.0.

Level	Value	Vulnerability	Risk (Required Action)
Critical	9 – 10	A vulnerability that can disrupt the program functioning in a number of scenarios, or creates a risk that the program may be broken.	Immediate action to reduce risk level.
High	7 – 8.9	A vulnerability that affects the desired outcome when using a program, or provides the opportunity to use a program in an unintended way.	Implementation of corrective actions as soon as possible.
Medium	4 – 6.9	A vulnerability that could affect the desired outcome of executing the program in a specific scenario.	Implementation of corrective actions in a certain period.
Low	2 – 3.9	A vulnerability that does not have a significant impact on possible scenarios for the use of the program and is probably subjective.	Implementation of certain corrective actions or accepting the risk.
Informational	0 – 1.9	A vulnerability that have informational character but is not effecting any of the code.	An observation that does not determine a level of risk

4. Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices. To do so, reviewed line-by-line by our team of expert pentesters and rust program developers, documenting any issues as there were discovered.

4.1 Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
 - i. Review of the specifications, sources, and instructions provided to Chainsulting to make sure we understand the size, scope, and functionality of the program
 - ii. Manual review of code, which is the process of reading source code line-by-line in an attempt to identify potential vulnerabilities.
 - iii. Comparison to specification, which is the process of checking whether the code does what the specifications, sources, and instructions provided to Chainsulting describe.
2. Testing and automated analysis that includes the following:
 - i. Test coverage analysis, which is the process of determining whether the test cases are actually covering the code and how much code is exercised when we run those test cases.
 - ii. Symbolic execution, which is analysing a program to determine what inputs causes each part of a program to execute.
3. Best practices review, which is a review of the programs to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, actionable recommendations to help you take steps to secure your programs.

5. Metrics

The metrics section should give the reader an overview on the size, quality, flows and capabilities of the codebase, without the knowledge to understand the actual code.

5.1 Tested Program Files

The following are the MD5 hashes of the reviewed files. A file with a different MD5 hash has been modified, intentionally or otherwise, after the security review. You are cautioned that a different MD5 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the review

File	Fingerprint (MD5)
./src/auction/cancel_auction_bid.rs	c96967feaf12a602d76c8a72146409f
./src/auction/cancel_auction.rs	bade2a7e7d280d7fff447e485129652e
./src/auction/claim_auction_revenue.rs	f8fdaccc648e312d7882fd3845fee7d8
./src/auction/mod.rs	6c5e9e03362083b06c99e84988202f42
./src/auction/state.rs	767d776cc2c3c2838caf7e6b01cdaeb2
./src/auction/make_bid.rs	b438695747ce27a03904bd4fa58380ed
./src/auction/create_auction.rs	9b379fc508647612d8d877c24d07cfc9
./src/auction/claim_lot_nft.rs	8f88fe07f0913ad5168a7fe6397ceb5d
./src/lib.rs	ad4d2def70c296af6c6f82ceb57040c8
./src/admin/add_currency_token_to_allowlist.rs	6ee9c114a5fcd23a964ce2cc3ebf4682
./src/admin/init_dusties_program.rs	7b8e8f2a65f896cecea340fb15d28f5b
./src/admin/add_nft_collection_to_allowlist.rs	5f1afce98b670f3847eb01b1c0d3f8b9
./src/admin/update_configs.rs	7e1689930da82324c72b703aa7417fc3
./src/admin/set_mock_timestamp.rs	e8bed1f91c1f41ed6dde70c591da9a44
./src/admin/mod.rs	dab1cc568a5a8ec91c356135c80b4721
./src/admin/state.rs	23a7da4730b27a5cd2b63fba59010d97
./src/common/raffle_strategy.rs	064fe8def28b9fa0d32e730a48cefb54

./src/common/types.rs	a88849539a01a0a9aaae68dac3666475
./src/common/revenue_distribution_strategy.rs	1a139de743f091c2e437b53409a14238
./src/common/random_utils.rs	0dda37ee1181e37fb0b60866c16a90ec
./src/common/bid_strategy.rs	91333806d9cf2d13848e935f9e4ca2c4
./src/common/auction_strategy.rs	00af4b56840424de6dfd27331a3e40ef
./src/common/error_code.rs	16951770299cccedc89b8f9f1ac9b591
./src/common/mod.rs	17545809b089d599d02db0fb8baee9c0
./src/common/pubkey_indexing.rs	44b81122bd245450784de4f1d112cdb0
./src/common/cpi_utils.rs	98af17006f22a3e0df651c1e3673392b
./src/common/eligibility_check_strategy.rs	3592d5c459d089669b62c0d60e862113
./src/common/utils.rs	d32271a9b097ccde0c844a9f238601f7
./src/common/configs_strategy.rs	7eafe074fc6b4807f3473a0c58ea5c3f
./src/raffle/claim_raffle_reward.rs	a2011f85c0bf3614c0f86b3303903c89
./src/raffle/create_raffle.rs	fe03fbc265b6e639625e8cb023877830
./src/raffle/mod.rs	6ecc6ad8d9599d7b7c2745af8fdd247d
./src/raffle/claim_raffle_revenue.rs	8bb00a343c1bc0b279b82c08dd3fbf21
./src/raffle/state.rs	56f6a4ce66db554b8c631842759443d3
./src/raffle/claim_remaining_raffle_rewards.rs	00c3971289b10bbb8af10946bc000855
./src/raffle/make_raffle.rs	f6b1c0d1c688fc4cedc55e773da1d85f
./src/raffle/buy_raffle_tickets.rs	b84212eadcd49f9e9854ea8ef15940b1
./src/raffle/cancel_raffle.rs	b5f5a548bc1e1f617c2406813b192854
./src/raffle/set_raffle_winners.rs	d25c38d51535dd96e36c1a1b7340255e

5.2 Used Code from other Frameworks/Dependencies

Dependency / Import Path	Source
getrandom	https://docs.rs/getrandom/0.2.8/getrandom/
anchor-lang	https://docs.rs/anchor-lang/0.26.0/anchor_lang/
anchor-spl	https://docs.rs/anchor-spl/0.26.0/anchor_spl/
mpl-token-metadata	https://docs.rs/mpl-token-metadata/1.4.3/mpl_token_metadata/
itertools	https://docs.rs/itertools/0.10.5/itertools/
rand	https://docs.rs/rand/0.8.5/rand/
rand_pcg	https://docs.rs/rand_pcg/0.3.1/rand_pcg/
bit-set	https://docs.rs/bit-set/0.5.3/bit_set/
wyhash	https://docs.rs/wyhash/0.5.0/wyhash/

5.3 Structure

Sort	Instruction	Functionality
Admin	init_dusties_program	
	update_configs	update_configs validate
	add_currency_to_allowlist	index_pubkey
	add_nft_collection_to_allowlist	index_pubkey
	set_mock_timestamp	
Auction	create_acution	init_auction desposit_nft
	cancel_auction	transfer_lot_nft close_lot_escrow_nft_account
	make_bid	validate_bid set_bid_account_data transfer_bid_funds make_bid
	cancel_bid	refund close_bid_escrow_token_account cancel_bid
	claim_lot_nft	transfer_lot_nft close_lot_escrow_nft_account finalize_auction_if_needd
	cliam_auction_revenue	validate distribute_revenue close_revenue_escrow_token_account finalize_auction_if_need
Raffle	create_raffle	init_raffle deposit_nft
	cancel_raffle	transfer_rewards try_close_rewards_escrow_nft_account

	close_revenue_escrow_token_account cancel
buy_raffle_tickets	try_index_ticket_position pay_for_tickets buy_tickets
make_raffle	validate_make_raffle make_raffle
set_raffle_winner	validate_set_winner set_winners
claim_raffle_reward	validate_claim claim_reward transfer_rewards try_close_rewards_escrow_nft_account
claim_raffle_revenue	validate distribute_revenue close_revenue_escrow_token_account
claim_remaining_raffle_rewards	validate_calim_remaining_rewards transfer_rewards try_close_rewards_escrow_nft_account

6. Scope of Work

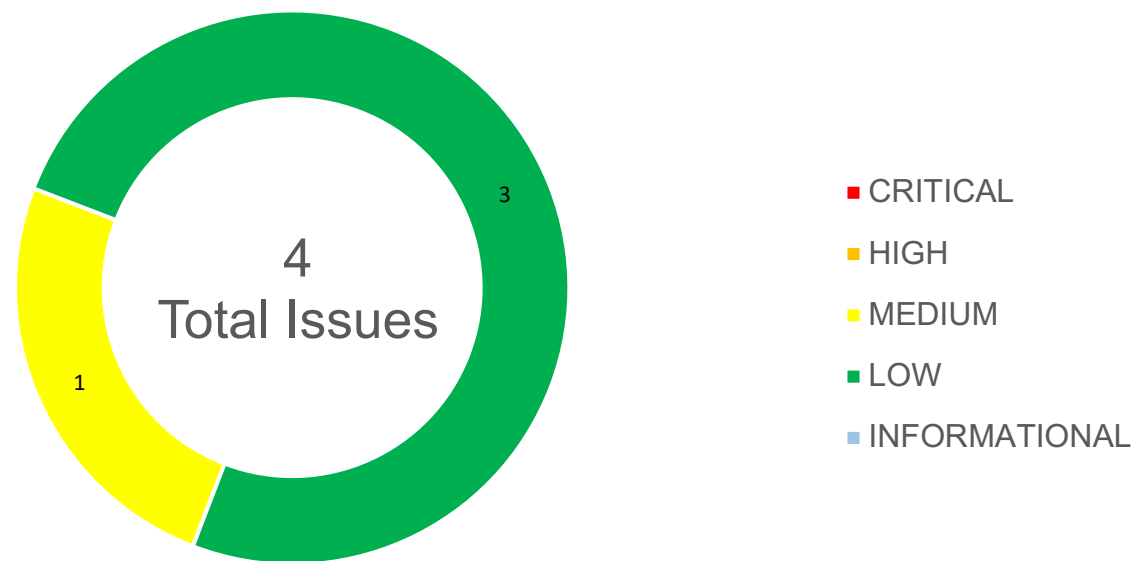
The Dust Labs developer team provided us with the files that needs to be tested. The scope of the audit is the dusties program.

The team put forward the following assumptions regarding the security, usage of the program:

- The program is coded according to the newest standards and in a secure way.
- The auctions are working as expected
- The raffles are working as expected
- The admin functions are working as expected

The main goal of this audit was to verify these claims. The auditors can provide additional feedback on the code upon the client's request.

6.1 Findings Overview



No	Title	Severity	Status
6.2.1	Integer Overflow and Underflow	MEDIUM	OPEN
6.2.2	Timestamp Dependencies	LOW	OPEN
6.2.3	Compiler Error	LOW	OPEN
6.2.4	Methods Execution Permissions (Access Control)	LOW	OPEN

6.2 Manual and Automated Vulnerability Test

CRITICAL ISSUES

During the audit, Chainsulting's experts found **0 Critical issues** in the code of the program.

HIGH ISSUES

During the audit, Chainsulting's experts found **0 High issues** in the code of the program.

MEDIUM ISSUES

During the audit, Chainsulting's experts found **1 Medium issue** in the code of the program.

6.2.1 Integer Overflow and Underflow

Severity: MEDIUM

Status: OPEN

Code: NA

File(s) affected: add_currency_token_to_allowlist.rs, add_nft_collection_to_allowlist.rs, auction_strategy.rs, bid_strategy.rs, raffle_strategy.rs, random_utils.rs, revenue_distribution.rs

Attack / Description	Do not use unchecked arithmetic operations. Unchecked operations can lead to integer overflow or underflow.
Code	programs/dusties/src/admin/add_currency_token_to_allowlist.rs : 102L programs/dusties/src/admin/add_nft_collection_to_allowlist.rs : 103L programs/dusties/src/common/auction_strategy.rs : 63L, 102-115L programs/dusties/src/common/bid_strategy.rs : 124L, 153L programs/dusties/src/common/raffle_strategy.rs : 167L, 176L, 237-247L, 334L programs/dusties/src/common/random_utils.rs : 11L, 32L, 44L programs/dusties/src/common/revenue_distribution.rs : 88L, 177L, 203L, 217-218L

Result/Recommendation	<p>It's recommended to use checked operations, that rustc can output the reason for unexpected errors when panicked on such arithmetic calculations. Please use checked_add() / checked_sub()</p> <p>https://doc.rust-lang.org/std/primitive.u64.html</p>
------------------------------	--

LOW ISSUES

During the audit, Chainsulting's experts found **3 Low issues** in the code of the program.

6.2.2 Timestamp Dependencies

Severity: LOW

Status: OPEN

Code: NA

File(s) affected: set_mock_timestamp.rs, utils.rs

Attack / Description	<p>Auctions and raffles use mock_timestamp which is normally used in the test environments. The value is set by the admin function - set_mock_timestamp. In the mainnet all programs use real time to get or set the data, so if the admin sets the mock timestamp manually on mainnet, all the raffles and auctions will be stopped or cause false behaviour. This functionality can cause problems once manipulated.</p>
Code	<p>Line 15 – 30 (set_mock_timestamp.rs)</p> <pre> #[derive(Accounts)] #[instruction(mock_timestamp: Option<i64>,)] pub struct SetMockTimestamp<'info> { #[account(mut, seeds = [b"global_states"], bump = global_states.bump, </pre>

	<pre> constraint = global_states.is_test_environment == true @DustiesErrorCode::NotTestEnvironment, constraint = global_states.authority == authority.key() @DustiesErrorCode::NotTheAuthority,)] pub global_states: Account<'info, GlobalStates>, pub authority: Signer<'info>, pub system_program: Program<'info, System>, } </pre>
Result/Recommendation	<p>If you want to retain this structure of the program, you can enable / disable the testing utilities by using the <code>#[cfg(feature)]</code> - “cfg expressions” provided by cargo.</p> <p>The admin function(<code>set_mock_timestamp</code>) and all other methods & traits for testing purposes should be removed from the codebase once deploying to mainnet (production). We have noticed the disclaimer/in-line comment within <code>set_mock_timestamp.rs</code> but see it as our duty to point out or remind about the possible issue.</p> <p>https://doc.rust-lang.org/cargo/reference/features.html</p>

6.2.3 Compiler Error

Severity: LOW

Status: OPEN

Code: NA

File(s) affected: types.rs

Attack / Description	Compile error occurred on rustc 1.67.0 / anchor 0.26.0 version.
Code	<pre>Line 27 - 69 (types.rs) error[E0658]: deriving `Default` on enums is experimental --> programs/dusties/src/common/types.rs:27:68 27 #[derive(AnchorSerialize, AnchorDeserialize, Clone, PartialEq, Eq, Default)] ^^^^^^^ = note: see issue #86985 <https://github.com/rust-lang/rust/issues/86985> for more information = help: add `#![feature(derive_default_enum)]` to the crate attributes to enable = note: this error originates in the derive macro `Default` (in Nightly builds, run with -Z macro-backtrace for more info) error[E0658]: deriving `Default` on enums is experimental --> programs/dusties/src/common/types.rs:69:67 69 AnchorSerialize, AnchorDeserialize, Clone, Copy, PartialEq, Eq, Default, ^^^^^^^ = note: see issue #86985 <https://github.com/rust-lang/rust/issues/86985> for more information = help: add `#![feature(derive_default_enum)]` to the crate attributes to enable</pre>

	<p>= note: this error originates in the derive macro `Default` (in Nightly builds, run with -Z macro-backtrace for more info)</p> <p>For more information about this error, try `rustc --explain E0658`. error: could not compile `dusties` due to 2 previous errors</p>
Result/Recommendation	Consider to use one of smooth crates, to use the default enum derivation feature. (smart-default, educe, defaults)

6.2.4 Methods Execution Permissions (Access Control)

Severity: LOW

Status: OPEN

Code: NA

File(s) affected: init_dusties_program.rs


Attack / Description	The global PDA initializer can still initialize, as long as the program deployer didn't init. Of course, it can be closed by the deployer (update_authority) but may need to change the global PDA seed, because we can't create a PDA on the same address which closed before.
Code	<p>Line 52 (init_dusties_program.rs)</p> <pre>// Please note that the authority is **immutable** after being initialized for // safety-reason. global_states.authority = ctx.accounts.authority.key();</pre>
Result/Recommendation	We recommend to load the program update authority programmatically and allow only the authority to init the global PDA.

INFORMATIONAL ISSUES


During the audit, Chainsulting's experts found **0 Informational issues** in the code of the program.

6.3 Verify Claims


6.3.1 The program is coded according to the newest standards and in a secure way.

Status: tested and verified 


6.3.2 The auctions are working as expected

Status: tested and verified 

6.3.3 The raffles are working as expected

Status: tested and verified 

6.3.4 The admin functions are working as expected

Status: tested and verified 

7. Executive Summary

Two (2) independent Chainsulting experts performed an unbiased and isolated audit of the solana program.

The main goal of the audit was to verify the claims regarding the security and functions of the program. During the audit, no critical, no high, 1 medium, 3 low and 0 informational issue have been found, after the manual and automated security testing.

We advise the Dust Labs team to implement the recommendations contained in all our findings, to further enhance the code's security and readability.

8. About the Auditor

Chainsulting is a professional software development firm, founded in 2017 and based in Germany. They show ways, opportunities, risks and offer comprehensive Web3 solutions. Their services include web3 development, security and consulting.

Chainsulting conducts code audits on market-leading blockchains such as Solana, Tezos, Ethereum, Binance Smart Chain, and Polygon to mitigate risk and instil trust and transparency into the vibrant crypto community. They have also reviewed and secure the smart contracts of many top DeFi projects.

Chainsulting currently secures [\\$100 billion](#) in user funds locked in multiple DeFi protocols. The team behind the leading audit firm relies on their robust technical know-how in the web3 sector to deliver top-notch smart contract audit solutions, tailored to the clients' evolving business needs.

Check our website for further information: <https://chainsulting.de>

How We Work



1 -----

PREPARATION

Supply our team with audit ready code and additional materials



2 -----

COMMUNICATION

We setup a real-time communication tool of your choice or communicate via e-mails.



3 -----

AUDIT

We conduct the audit, suggesting fixes to all vulnerabilities and help you to improve.



4 -----

FIXES

Your development team applies fixes while consulting with our auditors on their safety.



5 -----

REPORT

We check the applied fixes and deliver a full report on all steps done.