CONTRACT
WOLF

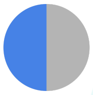
Security Assessment

Dropr Dapp

Verified on 03/14/2025

SUMMARY

Project	CHAIN	METHODOLOGY
Dropr	Solana	Manual & Automatic Analysis
FILES	DELIVERY	TYPE
Single	03/14/2025	Dapp Audit



Total Findings	Critical	Major	Medium	Minor	Informational	Resolved
4	0	0	0	2	2	0

🟥 0 Critical	An exposure that can affect the dapp's functions in several events that can risk and disrupt the code
🟧 0 Major	An opening & exposure to manipulate the code in an unwanted manner
🟧 0 Medium	An opening that could affect the outcome in executing the code in a specific situation
⬜ 2 Minor	An opening but doesn't have an impact on the functionality of the code
🟦 2 Informational	An opening that consists information but will not risk or affect the code
🟩 0 Resolved	ContractWolf's findings has been acknowledged & resolved by the project

STATUS ✔️**AUDIT PASSED**

TABLE OF CONTENTS | Dropr Dapp

Summary

Project Information

Findings

DISCLAIMER | Dropr Dapp

ContractWolf audits and reports should not be considered as a form of project's "Advertisement" and does not cover any interaction and assessment from "Project Code" to "External Code"

ContractWolf does not provide any warranty on its released report and should not be used as a decision to invest into audited projects.

ContractWolf provides a transparent report to all its "Clients" and to its "Clients Participants" and will not claim any guarantee of bug-free code within its **DAPP**.

ContractWolf's presence is to analyze, audit and assess the Client's Dapp to find any underlying risk and to eliminate any logic and flow errors within its code.

Each company or project should be liable to its security flaws and functionalities.

SCOPE OF WORK | Dropr Dapp

Dropr's team has agreed and provided us with the files that need to be tested. The scope of audit is the main dapp.

The goal of this engagement is to identify if there is a possibility of security flaws in the implementation of dapp and its systems.

ContractWolf will be focusing on dapp issues and functionalities along with the project claims from smart contract to their website, whitepaper, repository which has been provided by **Dropr**.

AUDITING APPROACH | Dropr Dapp

Every line of code along with its functionalities will undergo manual review to check for security issues, quality of logic and dapp scope of inheritance. The manual review will be done by our team that will document any issues that they discovered.

METHODOLOGY

The auditing process follows a routine series of steps :

1. Code review that includes the following :
   - Review of the specifications, sources and instructions provided to ContractWolf to make sure we understand the size, scope and functionality of the DAPP.
   - Manual review of code. Our team will have a process of reading the code line-by-line with the intention of identifying potential vulnerabilities, underlying and hidden security flaws.

2. Testing and automated analysis that includes :
   - Testing the DAPP’s function with common test cases and scenarios to ensure that it returns the expected results.

3. Best practices and ethical review. The team will review the dapp with the aim to improve efficiency, effectiveness, clarifications, maintainability, security and control within the dapp.

4. Recommendations to help the project take steps to eliminate or minimize threats and secure the dapp.

TOKEN DETAILS | Dropr Dapp

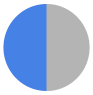
Dropr is the ultimate token airdrop platform created by BABA . Effortlessly distribute tokens to your community with automated claims, top-tier security, and real-time tracking.

Token Name	Symbol	Decimal	Total Supply	Chain
-	-	-	-	Solana

SOURCE

Source	*Sent Via local-files*

FINDINGS | Dropr Dapp



Total Findings	Critical	Major	Medium	Minor	Informational	Resolved
2	0	0	0	2	2	0

This report has been prepared to state the issues and vulnerabilities for Dropr Dapp through this audit. The goal of this report findings is to identify specifically and fix any underlying issues and errors

Dropr*(WalletClient)*

ID	Title	File & Line #	Severity	Status
DCW-012	Error Handling	L: 68, 64	Informational	● Pending
DCW-015	Potential Backdoor	L: 189	Informational	● Pending
DCW-016	Sensitive Data Exposure	L : 107, 140, 166, 192, 218, 244	Minor	● Pending
DCW-016	Sensitive Data Exposure	L: 295, 378	Minor	● Pending

PENETRATION ATTACKS | Dropr Dapp

Dapp Weakness Classification and Test Cases

ID	Description	Status
DCW-001	Malware Scan	● Passed
DCW-002	Phishing	● Passed
DCW-003	Missing HTTP Headers	● Passed
DCW-004	Valid SSL Certificate	● Passed
DCW-005	Firewalls(Drop & Deny)	● Passed
DCW-006	Potential SQL Injection	● Passed
DCW-007	Framework Version	● Passed
DCW-008	Gas Griefing	● Passed
DCW-009	Address Approval	● Passed
DCW-010	Address Draining	● Passed
DCW-011	Insecure API Usage	● Passed
DCW-012	Error Handling	● Informational
DCW-013	Memory Leak	● Passed
DCW-014	Lack of Input Validation	● Passed
DCW-015	Potential Backdoor	● Informational
DCW-016	Sensitive Data Exposure	● Minor
DCW-017	Request Limit	● Passed
DCW-018	Overflow or Precision Loss	● Passed
DCW-019	Unintended Behavior	● Passed

FIXES & RECOMMENDATION

DCW-012 | Error Handling

The `connect()` function assumes the access token remains valid indefinitely. If the token expires, all API calls will fail silently, returning null.

Risk :

Repeated failed requests could trigger rate *limits*, cause *downtime*, or expose the application to *API abuse patterns*.

Recommendation

Implement token expiration handling by checking `expires_in` or response errors. Automatically refresh tokens when needed.

DCW-012 | Potential Backdoor

Although this class handles *server-to-server* requests, if any endpoints are exposed (e.g., to webhooks or frontend), there's no **CSRF** prevention.

A **CSRF** attack could trick an authenticated user into performing unintended actions, like sending tokens.

Recommendation:

For public-facing routes :

Use Laravel’s CSRF protection (`@csrf in forms, csrf_token()` in API headers).

Require additional authentication (e.g., **PIN, 2FA**) for sensitive actions like transfers.

DCW-016 | Sensitive Data Exposure

The code frequently logs errors, including raw API responses:

```
Log::error($e->getMessage());
```

If an exception contains sensitive data (like API secrets or transaction details), this may leak into logs.

Recommendation

- Sanitize logs to avoid exposing request bodies or headers.
- Mask sensitive data ('*****') in **logs** (especially in production).
- Remove this error handling altogether.

DCW-016 | Sensitive Data Exposure

The code stores pincodes in plaintext (e.g., `$wallet->pincode`), which is a critical security risk.

Recommendation:

Encrypt pincodes at rest using a **secure encryption method** (e.g., *AES-256, SHA-256*).

Avoid logging or exposing pincodes in headers (e.g., signing-method header).

AUDIT COMMENTS | Dropr Dapp

Dapp audit comment for a non-technical perspective

🟢 Project has been marked as SAFE to be interacted with by any SVM wallets *(03-14-2025)*
🟢 DAPP has no backdoors
🟢 DAPP cannot drain wallets via approval

CONTRACTWOLF

Blockchain Security - Smart Contract Audits