PALADIN
BLOCKCHAIN SECURITY

Smart Contract Security Assessment

Final Report

For Avvy Domains (Resolvers)

09 July 2022

paladinsec.co

info@paladinsec.co

Table of Contents

Disclaimer

Paladin Blockchain Security ("Paladin") has conducted an independent audit to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the codes that were provided for the scope of this audit. This audit report does not constitute agreement, acceptance or advocation for the Project that was audited, and users relying on this audit report should not consider this as having any merit for financial advice in any shape, form or nature. The contracts audited do not account for any economic developments that may be pursued by the Project in question, and that the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are completely free of exploits, bugs, vulnerabilities or deprecation of technologies. Further, this audit report shall not be disclosed nor transmitted to any persons or parties on any objective, goal or justification without due written assent, acquiescence or approval by Paladin.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence regardless of the findings presented in this report. Information is provided 'as is', and Paladin is under no covenant to the completeness, accuracy or solidity of the contracts audited. In no event will Paladin or its partners, employees, agents or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions and/or actions with regards to the information provided in this audit report.

Cryptocurrencies and any technologies by extension directly or indirectly related to cryptocurrencies are highly volatile and speculative by nature. All reasonable due diligence and safeguards may yet be insufficient, and users should exercise considerable caution when participating in any shape or form in this nascent industry.

The audit report has made all reasonable attempts to provide clear and articulate recommendations to the Project team with respect to the rectification, amendment and/or revision of any highlighted issues, vulnerabilities or exploits within the contracts provided. It is the sole responsibility of the Project team to sufficiently test and perform checks, ensuring that the contracts are functioning as intended, specifically that the functions therein contained within said contracts have the desired intended effects, functionalities and outcomes of the Project team.

Paladin retains full rights over all intellectual property (including expertise and new attack or exploit vectors) discovered during the audit process. Paladin is therefore allowed and expected to re-use this knowledge in subsequent audits and to inform existing projects that may have similar vulnerabilities. Paladin may, at its discretion, claim bug bounties from third-parties while doing so.

1 Overview

This report has been prepared for Avvy Domains's Resolvers contracts on the Avalanche network. Paladin provides a user-centred examination of the smart contracts to look for vulnerabilities, logic errors or other issues from both an internal and external perspective.

1.1 Summary

Project Name	Avvy Domains
URL	https://avvy.domains/
Network	Avalanche
Language	Solidity

1.2 Contracts Assessed

Name	Contract	Live Code Match
PublicResolverV1	0x55f452383C0F0150CD440d077cDEE67de11B005d	MATCH
ResolverRegistryV1	0x3947d4c62C108A8A7bA3ED53AbaDcFF5D8998637	MATCH

1.3 Findings Summary

Severity	Found	Resolved	Partially Resolved	Acknowledged (no change made)
High	0	-	-	-
Medium	2	1	-	1
Low	2	-	-	2
Informational	9	-	-	9
Total	**13**	**1**	**-**	**12**

Classification of Issues

Severity	Description
High	Exploits, vulnerabilities or errors that will certainly or probabilistically lead towards loss of funds, control, or impairment of the contract and its functions. Issues under this classification are recommended to be fixed with utmost urgency.
Medium	Bugs or issues that may be subject to exploit, though their impact is somewhat limited. Issues under this classification are recommended to be fixed as soon as possible.
Low	Effects are minimal in isolation and do not pose a significant danger to the project or its users. Issues under this classification are recommended to be fixed nonetheless.
Informational	Consistency, syntax or style best practices. Generally pose a negligible level of risk, if any.

1.3.1 PublicResolverV1

ID	Severity	Summary	Status
01	MEDIUM	A suspended domain name still returns the previously stored `(resolver, datasetId)` pairs	ACKNOWLEDGED
02	LOW	Users might have difficulty knowing which paths/hashes have been set on their domain name	ACKNOWLEDGED
03	INFO	set and get can be made external	ACKNOWLEDGED
04	INFO	Gas optimizations	ACKNOWLEDGED
05	INFO	Unused import: PoseidonInterface.sol	ACKNOWLEDGED
06	INFO	Typographical error	ACKNOWLEDGED

1.3.2 ResolverRegistryV1

ID	Severity	Summary	Status
07	MEDIUM	A suspended domain name still returns the previously stored data	✓ RESOLVED
08	LOW	Users might have difficulties knowing which (path/hash, key) pairs have been set on their domain name	ACKNOWLEDGED
09	INFO	The `setStandard` function lacks validation on key	ACKNOWLEDGED
10	INFO	Various functions can be made external	ACKNOWLEDGED
11	INFO	Gas optimizations	ACKNOWLEDGED
12	INFO	Unused import: PoseidonInterface.sol	ACKNOWLEDGED
13	INFO	Typographical errors	ACKNOWLEDGED

2 Findings

2.1 PublicResolverV1

`PublicResolverV1` allows users to set a resolver for their domain name. A subdomain can link to a different domain than the domain itself. In this case, the user needs to provide the subdomain as the `path` parameter.

Only the domain's owner can call `set` to set a resolver and a `datasetId` on their domain name. If a path is provided, it will be hashed by the `rainbowTable` contract that was audited in the previous audit. If no path is provided, the hash will be the name itself.

To reset a (sub)domain, the owner must set it back to `address(0)`.

At the time of the audit, there is only one resolver, `PublicResolverV1`, that is audited below this one.

2.1.1 Privileged Functions

- `set [only domain owner]`

2.1.2 Issues & Recommendations

Issue #01	**A suspended domain name still returns the previously stored (`resolver`, `datasetId`) pairs**
Severity	MEDIUM SEVERITY
Description	On suspension, a domain cannot be transferred or set anymore. However, the previously set (`resolver`, `datasetId`) pairs are still returned when querying the right hash as it does not check for the domain's validity.
Recommendation	Consider reverting to a suspended domain if this behavior isn't wanted.
Resolution	ACKNOWLEDGED

Issue #02	**Users might have difficulty knowing which paths/hashes have been set on their domain name**
Severity	LOW SEVERITY
Description	When a domain is transferred (because it expired or was sent by the owner), the previously stored values are not reset and are still returned. As it is impossible to know which hash was set without looking at events, users may be unaware if a hidden value remains on their newly acquired domain or if they set it a long ago and forgot the path/hash.
Recommendation	Consider using OpenZepellin's `enumerableSet` instead of mapping to allow users to easily see which paths/hashes have been set on their domain.
Resolution	ACKNOWLEDGED The client has indicated that the users would need to use events for that matter. Avvy will provide tooling to generate those off-chain representations.

Issue #03	**`set` and `get` can be made `external`**
Severity	INFORMATIONAL
Description	Functions that are not used within the contract but only externally can be marked as such with the `external` keyword. Apart from being a best practice when the function is not used within the contract, this can lead to lower gas usage in certain cases.
Recommendation	Consider marking the functions mentioned above as `external`.
Resolution	ACKNOWLEDGED

Issue #04	Gas optimizations
Severity	INFORMATIONAL

Description	The contract contains multiple code sections that could be further optimized for gas efficiency. We have consolidated these in a single issue in an effort to keep the report brief and readable.
	
	L10
	`ContractRegistryInterface contractRegistry;`
	
	Consider marking `contractRegistry` as `immutable` to save some gas as it will hardcode the value of the bytecode during deployment. The variable should also be marked as `public` so users can inspect it easily.
	
	L36
	`require(resolvers[name][hash] != address(0), "ResolverRegistry: resolver not set");`
	
	L37
	`return (resolvers[name][hash], datasetIds[name][hash]);`
	
	Consider caching the `resolver` returned by the mapping to save some gas as it is unnecessary to load it twice from storage.
	
	`calldata` can be used throughout the contract to save on gas.

Recommendation	Consider implementing the gas optimizations mentioned above.
Resolution	ACKNOWLEDGED

Issue #05	Unused import: PoseidonInterface.sol
Severity	INFORMATIONAL
Location	L5 `import "./PoseidonInterface.sol";`
Description	Files imported in a contract but not used within said contract could confuse third-party auditors. They also increase the contract length unnecessarily.
Recommendation	Consider removing the import to keep the contract short and simple.
Resolution	ACKNOWLEDGED

Issue #06	Typographical error
Severity	INFORMATIONAL
Location	L49 `constructor(address contractRegistryAddress) {`
Description	`contractRegistryAddress` could be casted to the right type directly.
Recommendation	Consider fixing the typographical error.
Resolution	ACKNOWLEDGED

2.2 ResolverRegistryV1

`ResolverRegistryV1` allows users to set data on their domain name or subdomain. Each domain or subdomain can link to different data depending on the different inputs. Each domain has 2 types of inputs: one standard type (that can be seen at the end of this description) and one for custom entries where the user can use the values they want.

Resolvers allow users to point their domain and subdomains to IP addresses, other domain names, wallet addresses, avatar images or even IPFS content hashes. They therefore allow domain owners to link their domain to various destinations.

Only the domain's owner can call `set` or `setStandard` to set data on their domain. If a user wants to use a subdomain, they need to use the `path` parameter. If a path is provided, it will be hashed by the `RaindowTable` contract that was audited during a previous audit by Paladin. If no path is provided, the hash will be the name itself.

To reset the data on a given domain, the user needs to use `set` with the right (`datasetId`, `path`, `key`) pair and an empty string.

The standard inputs are as of the time of this audit listed below (it should be noted that the regex part has not been filled out yet):

Key	Name	Label	Description
1	X_CHAIN	Address on Avalanche X-Chain	Address on Avalanche X-Chain
2	P_CHAIN	Address on Avalanche P-Chain	Address on Avalanche P-Chain
3	EVM	C-Chain / EVM Address	Address on EVM-type network, including Avalanche C-Chain
4	VALIDATOR	Validator NodeID	Validator NodeID on the Avalanche Network
5	DNS_CNAME	DNS CNAME Record	DNS CNAME Record
6	DNS_A	DNS A Record	DNS A Record
7	AVATAR	Avatar	An image which the user wishes to use as their avatar. Value should be a URL which references the image.
8	CONTENT	Content	A downloadable file. Value should be a URL (e.g. IPFS, HTTPS, ..) which references the image.

2.2.1 Privileged Functions

- `setStandard [only domain owner]`
- `set [only domain owner]`

2.2.2 Issues & Recommendations

Issue #07	A suspended domain name still returns the previously stored data
Severity	MEDIUM SEVERITY
Description	Once suspended, a domain cannot be transferred or set anymore. However, the previously set data are still returned when querying the right (hash, key) pair as it does not check for the domain's validity.
Recommendation	Consider reverting to a suspended domain if this behavior is not desired on `resolveStandard` and `resolve`.
Resolution	RESOLVED The client has indicated that even if this contract reverted, users could still use their own smart contract to avoid the suspension. Additionally, reverting within the PublicResolver is the only effective way to prevent users from using a suspended domain name. This issue was resolved as it turned out to be desired behavior. Users should however understand that the behavior did not change and the issue description is therefore still present.

Issue #08	**Users might have difficulties knowing which (path/hash, key) pairs have been set on their domain name**
Severity	LOW SEVERITY
Description	When a domain is transferred (because it expired or was sent by the owner), the previously stored values are not reset and are still returned. As it is impossible to know which `(hash, key)` were set without looking at events, users may be unaware if a hidden value remains on their newly acquired domain or if they set it a long time ago and forgot the path/hash.
Recommendation	Consider using OpenZepellin's `enumerableSet` instead of mapping to allow users to easily see which (path/hash, key) have been set on their domain.
Resolution	ACKNOWLEDGED The client has indicated that the users would need to use events for that matter. Avvy will provide tooling to generate those off-chain representations.

Issue #09	**The `setStandard` function lacks validation on key**
Severity	INFORMATIONAL
Description	The `setStandard` function should check that the `key` parameter is within the bounds of the standard inputs to avoid users setting a key that is not a `standardInput` yet and in the future is set to one that will probably not match the standard.
Recommendation	Consider validating the `key` parameter on the `setStandard` function.
Resolution	ACKNOWLEDGED

Issue #10	Various functions can be made external
Severity	INFORMATIONAL
Description	Functions that are not used within the contract but only externally can be marked as such with the external keyword. Apart from being a best practice when the function is not used within the contract, this can lead to a lower gas usage in certain cases. The following functions can be marked as external: - `resolveStandard` - `setStandard` - `resolve` - `set`
Recommendation	Consider marking the functions mentioned above as external.
Resolution	ACKNOWLEDGED

Issue #11	Gas optimizations
Severity	INFORMATIONAL
Description	The contract contains multiple code sections that could be further optimized for gas efficiency. We have consolidated these in a single issue in an effort to keep the report brief and readable. L10 `ContractRegistryInterface contractRegistry;` Consider marking `contractRegistry` as `immutable` to save some gas as it will hardcode the value of the bytecode during deployment. The variable should also be marked as `public` so users can inspect it easily. `calldata` can be used throughout the contract to save on gas.
Recommendation	Consider implementing the gas optimizations mentioned above.
Resolution	ACKNOWLEDGED

Issue #12	Unused import: PoseidonInterface.sol
Severity	INFORMATIONAL
Location	L5 `import "./PoseidonInterface.sol";`
Description	Files imported in a contract but not used within said contract could confuse third-party auditors. They also increase the contract length unnecessarily.
Recommendation	Consider removing the import to keep the contract short and simple.
Resolution	ACKNOWLEDGED

Issue #13	Typographical errors
Severity	INFORMATIONAL

Description	The contract contains a number of typographical errors which we have consolidated below in a single issue in an effort to keep the report size reasonable.
	
	L44
	`require(!domain.isSuspended(name), "ResolverRegistry: domain suspended");`
	
	The comment should mention `ResolverV1` instead of `ResolverRegistry`.
	
	L55
	`constructor(address contractRegistryAddress) {`
	
	`contractRegistryAddress` could be casted to the right type directly.

Recommendation	Consider fixing the typographical errors.
Resolution	ACKNOWLEDGED


PALADIN
BLOCKCHAIN SECURITY