



# SOLIDProof

*Bring trust into your projects*

**Blockchain Security | Smart Contract Audits | KYC  
Development | Marketing**

MADE IN GERMANY

## JURISPROTOCOL

# AUDIT

SECURITY ASSESSMENT

**30. September, 2024**

FOR



[SolidProof.io](https://solidproof.io)



[@solidproof\\_io](https://t.me/solidproof_io)



Introduction	3
Disclaimer	3
Project Overview	4
Summary	4
Social Medias	4
Audit Summary	5
File Overview	5
Imported packages	5
Audit Information	6
Vulnerability & Risk Level	6
Auditing Strategy and Techniques Applied	7
Methodology	7
Overall Security	8
Upgradeability	8
Ownership	9
Ownership Privileges	10
Minting tokens	10
Burning tokens	11
Blacklist addresses	12
Fees and Tax	13
Lock User Funds	14
Centralization Privileges	15
Audit Results	16



## Introduction

[SolidProof.io](https://solidproof.io) is a brand of the officially registered company FutureVisions Deutschland, based in Germany. We're mainly focused on Blockchain Security such as Smart Contract Audits and KYC verification for project teams.

Solidproof.io assess potential security issues in the smart contracts implementations, review for potential inconsistencies between the code base and the whitepaper/documentation, and provide suggestions for improvement.

## Disclaimer

[SolidProof.io](https://solidproof.io) reports are not, nor should be considered, an "endorsement" or "disapproval" of any particular project or team. These reports are not, nor should be considered, an indication of the economics or value of any "product" or "asset" created by any team. SolidProof.io do not cover testing or auditing the integration with external contract or services (such as Unicrypt, Uniswap, PancakeSwap etc'...)

**SolidProof.io Audits do not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technology proprietors. SolidProof Audits should not be used in any way to make decisions around investment or involvement with any particular project. These reports in no way provide investment advice, nor should be leveraged as investment advice of any sort.**

SolidProof.io Reports represent an extensive auditing 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. SolidProof's position is that each company and individual are responsible for their own due diligence and continuous security. SolidProof in no way claims any guarantee of the security or functionality of the technology we agree to analyze.

# Project Overview

## Summary

Project Name	Juris Protocol
Website	TBA
About the project	While \$LUNC has built a strong community and shows impressive liquidity, it faces a significant challenge: the lack of founders and practical uses. This gap has limited the ecosystem's ability to fully leverage its blockchain technology. <a href="#">Juris Protocol</a> is stepping in to fill this void by bringing in visionary ideas and creating valuable applications.
Chain	Terra Classic
Language	Rust (Token-Program)
Codebase	Provided by file
Commit	N/A
Unit Tests	Not Provided

## Social Medias

Telegram	<a href="https://t.me/jurisprotocol">https://t.me/jurisprotocol</a>
Twitter	<a href="https://x.com/JurisProtocol">https://x.com/JurisProtocol</a>
Facebook	N/A
Instagram	N/A
GitHub	<a href="https://github.com/JurisProtocol">https://github.com/JurisProtocol</a>
Reddit	<a href="https://www.reddit.com/r/JurisProtocol/">https://www.reddit.com/r/JurisProtocol/</a>
Medium	<a href="https://medium.com/@jurisprotocol">https://medium.com/@jurisprotocol</a>
Discord	<a href="https://discord.com/invite/FzQwHYgjAd">https://discord.com/invite/FzQwHYgjAd</a>
YouTube	<a href="https://www.youtube.com/@JurisProtocol">https://www.youtube.com/@JurisProtocol</a>
TikTok	N/A
LinkedIn	N/A

## Audit Summary

Version	Delivery Date	Change Log
v1.0	30. September 2024	<ul style="list-style-type: none"> <li>Layout Project</li> <li>Automated/Manual-Security Testing</li> <li>Summary</li> </ul>

**Note** – The following audit report presents a comprehensive security analysis of the smart contract utilized in the project that includes outside manipulation of the contract's functions in a malicious way. This analysis did not include functional testing (or unit testing) of the contract/s logic. We cannot guarantee 100% logical correctness of the contract as we did not functionally test it. This includes internal calculations in the formulae used in the contract.

## File Overview

The Team provided us with the files that should be tested in the security assessment. This audit covered the following files listed below with a SHA-1 Hash.

### 1. Cw-20-taxed01.1.0-taxed003



2. please note: Files with a different hash value than in this table have been modified after the security check, either intentionally or unintentionally. A different hash value may (but need not) be an indication of a changed state or potential vulnerability that was not the subject of this scan.



## Imported packages

Used code from other Frameworks/Smart Contracts (direct imports).

1. contract
2. cosmwasm-schema = "1.1.0"
3. cw2 = "1.1.2"
4. cw20 = "1.1.2"
5. cw20-base = "1.1.2"
6. cw-storage-plus = "1.1.2"
7. cosmwasm-std = "1.1.0"
8. schemars = "0.8.1"
9. semver = "1"
10. serde = "1.0.103"
11. thiserror = "1.0.23"
12. cw-multi-test = "1.0.2"
13. cw-utils = "1.0.2"

Please note: Files with a different hash value than in this table have been modified after the security check, either intentionally or unintentionally. A different hash value may (but need not) be an indication of a changed state or potential vulnerability that was not the subject of this scan.

## Audit Information

### Vulnerability & Risk Level

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

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

## Auditing Strategy and Techniques Applied

Throughout the review process, care was taken to check the repository for security-related issues, code quality, and compliance with specifications and best practices. To this end, our team of experienced pen-testers and smart contract developers reviewed the code line by line and documented any issues discovered.

We check every file manually. We use automated tools only so that they help us achieve faster and better results.

## Methodology

The auditing process follows a routine series of steps:

1. Code review that includes the following:
  - a. Reviewing the specifications, sources, and instructions provided to SolidProof to ensure we understand the size, scope, and functionality of the smart contract.
  - b. Manual review of the code, i.e., reading the source code line by line to identify potential vulnerabilities.
  - c. Comparison to the specification, i.e., verifying that the code does what is described in the specifications, sources, and instructions provided to SolidProof.
2. Testing and automated analysis that includes the following:
  - a. Test coverage analysis determines whether test cases cover code and how much code is executed when those test cases are executed.
  - b. Symbolic execution, which is analysing a program to determine what inputs cause each part of a program to execute.
3. Review best practices, i.e., review smart contracts to improve efficiency, effectiveness, clarity, maintainability, security, and control based on best practices, recommendations, and research from industry and academia.
4. Concrete, itemized and actionable recommendations to help you secure your smart contracts.





## Overall Security Upgradeability

**Contract is not an upgradeable**



**Deployer cannot update the contract with new functionalities.**

Description

The contract is not an upgradeable contract. The Deployer is able to change or add any functionalities to the contract after deploying.

Comment

N/A



## Ownership

**Contract ownership is renounced.**

 **The ownership is renounced.**

Description

The owner has renounced the ownership which means that the owner retains control over the contract's operations, including the ability to execute functions that may impact the contract's users or stakeholders. This can lead to several potential issues, including:

- Centralizations
- The owner has significant control over the contract's operations.

Comment

N/A

**Note** – *The contract cannot be considered as renounced till it is not deployed or having some functionality that can change the state of the contract.*



## Ownership Privileges

*These functions can be dangerous. Please note that abuse can lead to financial loss. We have a guide where you can learn more about these Functions.*

## Minting tokens

*Minting tokens refer to the process of creating new tokens in a cryptocurrency or blockchain network. This process is typically performed by the project's owner or designated authority, who has the ability to add new tokens to the network's total supply.*

**Contract owner can mint new tokens.**

**✗ The owner can mint new tokens.**

Description

The owner is able to mint new tokens once the contract is deployed.

Comment

The TOKEN\_INFO minter can mint new tokens to the recipient address.



## Burning tokens

*Burning tokens is the process of permanently destroying a certain number of tokens, reducing the total supply of a cryptocurrency or token. This is usually done to increase the value of the remaining tokens, as the reduced supply can create scarcity and potentially drive up demand.*

**Contract owner can burn tokens**

**✗ The owner can burn tokens.**

Description

The token holders can burn their tokens.

Comment

NA



## Blacklist addresses

*Blacklisting addresses in smart contracts is the process of adding a certain address to a blacklist, effectively preventing them from accessing or participating in certain functionalities or transactions within the contract. This can be useful in preventing fraudulent or malicious activities, such as hacking attempts or money laundering.*

**Contract owner cannot blacklist addresses.**



**The owner cannot blacklist wallets.**

Description	The owner is not able to blacklist addresses to lock funds.
Comment	N/A



## Fees and Tax

*In some smart contracts, the owner or creator of the contract can set fees for certain actions or operations within the contract. These fees can be used to cover the cost of running the contract, such as paying for gas fees or compensating the contract's owner for their time and effort in developing and maintaining the contract.*

**Contract owner cannot set fees more than 25%.**



**The owner cannot set fees more than 25%.**

Description	The owner cannot set fees of more than 25%.
Comment	N/A



## Lock User Funds

*In a smart contract, locking refers to the process of restricting access to certain tokens or assets for a specified period of time. When token or assets are locked in a smart contract, they cannot be transferred or used until the lock-up period has expired or certain conditions have been met.*

**Contract owner cannot lock function.**

☒ **The owner cannot lock function.**

Description	The owner cannot lock the contract.
Comment	N/A

## Centralization Privileges

*Centralization can arise when one or more parties have privileged access or control over the contract's functionality, data, or decision-making. This can occur, for example, if the contract is controlled by a single entity or if certain participants have special permissions or abilities that others do not.*

In the project, there are authorities that have access to the following functions:

File	Privileges
<b>Contract.rs</b>	<ul style="list-style-type: none"> <li>➤ The TAX_INFO admin can set tax map info.</li> <li>➤ The TAX_INFO admin can set tax map admin.</li> <li>➤ The TOKEN_INFO minter can mint new tokens to the recipient's address.</li> <li>➤ The TOKEN_INFO minter can update MinterData.</li> <li>➤ The MARKETING_INFO admin can update marketing info.</li> <li>➤ The MARKETING_INFO admin can update logo info.</li> </ul>

## Recommendations

To avoid potential hacking risks, it is advisable for the client to manage the private key of the privileged account with care. Additionally, we recommend enhancing the security practices of centralized privileges or roles in the protocol through a decentralized mechanism or smart-contract-based accounts, such as multi-signature wallets.

Here are some suggestions of what the client can do:

- Consider using multi-signature wallets: Multi-signature wallets require multiple parties to sign off on a transaction before it can be executed, providing an extra layer of security e.g. Gnosis Safe
- Use of a timelock at least with a latency of e.g. 48-72 hours for awareness of privileged operations
- Introduce a DAO/Governance/Voting module to increase transparency and user involvement
- Consider Renouncing the ownership so that the owner cannot modify any state variables of the contract anymore. Make sure to set up everything before renouncing.



## Audit Result

### Critical Issues

No critical issues

### High Issues

No high issues

### Medium Issue

#### #1 | Missing validation check.

File	Severity	Location	Status
Contract.rs	Medium	L433	Open

**Description** – Missing validation check for the recipient address is not a whale after minting tokens in execute\_mint function. It is recommended to add a validation check for the whale account after minting tokens on the recipient address.

#### #2 | Missing validation check.

File	Severity	Location	Status
Contract.rs	Medium	L287	Open

**Description** – Missing validation check of max\_tax for setting new tax map info in execute\_set\_tax\_map function . It is recommended to add a validation check of max tax amount for setting tax info for transfers in execute\_set\_tax\_map function.

#### #3 | Missing validation check.

File	Severity	Location	Status
Contract.rs	Medium	L769	Open

**Description** – Missing validation check of max\_tax for setting new tax map info in migrate function. It is recommended to add a validation check of max tax amount for setting tax info for transfers in migrate function.

## Low Issue

---

### #1 | Missing zero amount check.

File	Severity	Location	Status
Contract.rs	Low	L332-,L478	Open

**Description** – Missing zero amount check in execute\_transfer/execute\_send function. It is recommended to add a zero amount validation check in execute\_transfer/execute\_send function. If the amount is 0, it should revert.

## Informational Issue

---

### #1 | Skip updating balance for tax.

File	Severity	Location	Status
Contract.rs	Informational	L360-, L378	Open

**Description** – Skip updating balance for tax if the recipient address is the proceeds wallet or tax is zero. Updating contract tax balance in case of zero tax if the recipient address is proceeds wallet can be avoided.



## Legend for the Issue Status

Attribute or Symbol	Meaning
<b>Open</b>	The issue is not fixed by the project team.
<b>Fixed</b>	The issue is fixed by the project team.
<b>Acknowledged(ACK)</b>	The issue has been acknowledged or declared as part of business logic.





**Blockchain Security | Smart Contract Audits | KYC  
Development | Marketing**

MADE IN GERMANY