Quantstamp Contract Security Certificate

November 25th 2019 — Quantstamp Verified

Sablier

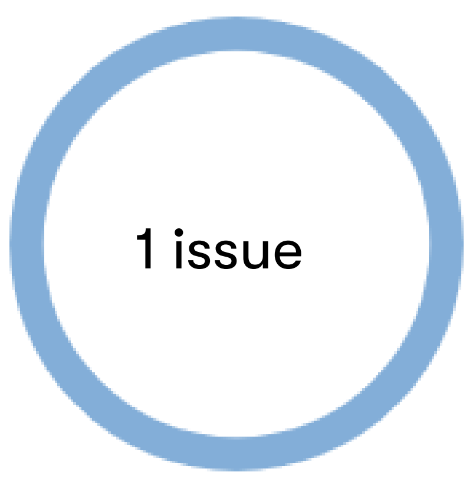
This smart contract audit was prepared by Quantstamp, the protocol for securing smart contracts.

Executive Summary

Type	Continuous Salary Smart Contract Audit
Auditors	Ed Zulkoski, Senior Security Engineer Kacper Bąk, Senior Research Engineer
Timeline	2019-10-16 through 2019-11-01
EVM	Constantinople
Languages	Solidity, Javascript
Methods	Architecture Review, Unit Testing, Functional Testing, Computer-Aided Verification, Manual Review
Specification	ERC-1620
Source Code	

Repository	Commit
sablier (initial report)	fc54b02
sablier (final report)	b415ea3

Total Issues	1 (1 Resolved)
High Risk Issues	0
Medium Risk Issues	0
Low Risk Issues	0
Informational Risk Issues	1 (1 Resolved)
Undetermined Risk Issues	0



Overall Assessment

The Sablier smart contracts are of overall high-quality, well documented, and thoroughly tested. No significant issues were found during the audit, and there were few gaps in the code-coverage provided by the test suite. We noted some minor discrepancies between the code and the specification which may simply require an update to the documents.

Update: Sablier has addressed our findings as of commit b415ea3.

Severity Categories	
⌃⌃ High	The issue puts a large number of users' sensitive information at risk, or is reasonably likely to lead to catastrophic impact for client's reputation or serious financial implications for client and users.
⌃ Medium	The issue puts a subset of users' sensitive information at risk, would be detrimental for the client's reputation if exploited, or is reasonably likely to lead to moderate financial impact.
⌄ Low	The risk is relatively small and could not be exploited on a recurring basis, or is a risk that the client has indicated is low-impact in view of the client's business circumstances.
○ Informational	The issue does not post an immediate risk, but is relevant to security best practices or Defence in Depth.
– Undetermined	The impact of the issue is uncertain.

Summary of Findings

ID	Description	Severity	Status
QSP-1	The struct `Stream` does not strictly adhere to the specification	○ Informational	Fixed

Goals

- Can funds be locked or stolen from the contracts?
- Are interest rates for compounding stream computed properly?
- Is access control properly limited to the relevant parties?

Changelog

- 2019-10-31 - Initial report
- 2019-11-01 - Final report

Quantstamp Audit Breakdown

Quantstamp's objective was to evaluate the repository for security-related issues, code quality, and adherence to specification and best practices.

Methodology

The Quantstamp 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 Quantstamp to make sure we understand the size, scope, and functionality of the smart contract
   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 Quantstamp 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 analyzing a program to determine what inputs cause each part of a program to execute.
3. Best practices review, which is a review of the smart contracts to improve efficiency, effectiveness, clarify, maintainability, security, and control based on the established industry and academic practices, recommendations, and research.
4. Specific, itemized, and actionable recommendations to help you take steps to secure your smart contracts.

Toolset

The notes below outline the setup and steps performed in the process of this audit.

Setup

Tool Setup:

- Maian
- Truffle
- Ganache
- SolidityCoverage
- Mythril
- Securify
- Slither

Steps taken to run the tools:

1. Installed Truffle: `npm install -g truffle`
2. Installed Ganache: `npm install -g ganache-cli`
3. Installed the solidity-coverage tool (within the project's root directory): `npm install --save-dev solidity-coverage`
4. Ran the coverage tool from the project's root directory: `./node_modules/.bin/solidity-coverage`
5. Installed the Mythril tool from Pypi: `pip3 install mythril`
6. Ran the Mythril tool on each contract: `myth -x path/to/contract`
7. Ran the Securify tool: `java -Xmx6048m -jar securify-0.1.jar -fs contract.sol`
8. Cloned the MAIAN tool: `git clone --depth 1 https://github.com/MAIAN-tool/MAIAN.git maian`
9. Ran the MAIAN tool on each contract: `cd maian/tool/ && python3 maian.py -s path/to/contract contract.sol`
10. Installed the Slither tool: `pip install slither-analyzer`
11. Run Slither from the project directory `slither .`

Assessment

Findings

QSP-1 The struct `Stream` does not strictly adhere to the specification

Severity: *Informational*

Status: Fixed

Contract(s) affected: `Types.sol`

Description: In particular, a new `isEntity` field has been added, and the order of the fields differs from the specification.

Recommendation: Consider drafting an updated version of the specification to account for these changes.

Automated Analyses

Maian

Maian did not detect any issues.

Mythril

Mythril detected several calls to user-supplied addresses, particularly for ERC20 transfers, as well as several calls to fixed addresses. This is discussed further in the Slither section below.

Securify

Securify reported several "Locked Ether" warnings, however each of these were associated irrelevant lines of code (e.g., blank lines or comment blocks), and as such were labelled as false positives.

Slither

Slither reported several potential reentrancy issues, however these external calls were to Sablier-owned contracts and therefore false positives. While there are several external calls to user-inputted external addresses of ERC20 tokens, such as in `Payroll.createSalary()`, this does not have any detrimental effects on other streams or the overall system. Further, if, for example, a stream is created using a malicious token, it is up to the recipient to ensure the validity of the underlying token.

Adherence to Specification

Update: fixed. The code adheres to the provided specification except for some very minor discrepancies, as noted above.

Code Documentation

The documentation is thorough.

Adherence to Best Practices

The code adheres to best practices. We suggest a few minor changes as follows:

- As already made aware in https://forum.openzeppelin.com/t/contract-request-ownable-without-renounceownership/1400, although the `transferOwnership()` function has been removed, the ownership can be *effectively* renounced if ownership is transferred to an unclaimed address. Consider incorporating the `Claimable` library when it becomes a standard component of OpenZeppelin.
- **Update: fixed.** The constructor of `Sablier` should check that the `cTokenManagerAddress != address(0)`.
- **Update: fixed.** In `Sablier.sol` on L541, line 541 "sender" should be "recipient".
- **Update: fixed.** Spelling: "blance", "timestmap", "remainig".

Test Results

Test Suite Results

```
yarn test
yarn run v1.17.3
$ yarn wsrun --package $PKG --serial -c test
$ wsrun --exclude-missing --fast-exit --package --serial -c test
@sablier/eslint-config has no test script, skipping missing
@sablier/shared-contracts has no test script, skipping missing
@sablier/burner-wallet has no test script, skipping missing
@sablier/dev-utils has no test script, skipping missing
@sablier/landing has no test script, skipping missing
@sablier/protocol
$ packages/protocol/scripts/test.sh
 | Starting our own ganache instance
 | Waiting for ganache to launch on port 8545...
 | You can improve web3's peformance when running Node.js versions older than 10.5.0 by installing
the (deprecated) scrypt package in your project
 | Connection to localhost port 8545 [tcp/*] succeeded!
 | Ganache launched!
$ /Users/ezulkosk/audits/sab2/node_modules/.bin/truffle version
 | You can improve web3's peformance when running Node.js versions older than 10.5.0 by installing
the (deprecated) scrypt package in your project
 | You can improve web3's peformance when running Node.js versions older than 10.5.0 by installing
the (deprecated) scrypt package in your project
 | Truffle v5.0.35 (core: 5.0.35)
 | Solidity - 0.5.11 (solc-js)
 | Node v8.11.4
 | Web3.js v1.2.1
$ /Users/ezulkosk/audits/sab2/node_modules/.bin/truffle test
 | You can improve web3's peformance when running Node.js versions older than 10.5.0 by installing
the (deprecated) scrypt package in your project
 | You can improve web3's peformance when running Node.js versions older than 10.5.0 by installing
the (deprecated) scrypt package in your project
 |
 | Compiling your contracts...
 | ===========================
 | > Compiling packages/protocol/contracts/CTokenManager.sol
 | > Compiling packages/protocol/contracts/Migrations.sol
 | > Compiling packages/protocol/contracts/Sablier.sol
 | > Compiling packages/protocol/contracts/Types.sol
 | > Compiling packages/protocol/contracts/interfaces/ICTokenManager.sol
 | > Compiling packages/protocol/contracts/interfaces/IERC1620.sol
 | > Compiling packages/protocol/contracts/test/Imports.sol
 | > Compiling @openzeppelin/contracts-packages/protocol/ethereum-package/contracts/GSN/Context.sol
 | > Compiling @openzeppelin/contracts-packages/protocol/ethereum-package/contracts/access/Roles.sol
 | > Compiling @openzeppelin/contracts-packages/protocol/ethereum-
package/contracts/access/roles/MinterRole.sol
 | > Compiling @openzeppelin/contracts-packages/protocol/ethereum-package/contracts/math/SafeMath.sol
 | > Compiling @openzeppelin/contracts-packages/protocol/ethereum-
package/contracts/token/ERC20/ERC20.sol
 | > Compiling @openzeppelin/contracts-packages/protocol/ethereum-
package/contracts/token/ERC20/ERC20Mintable.sol
 | > Compiling @openzeppelin/contracts-packages/protocol/ethereum-
package/contracts/token/ERC20/IERC20.sol
 | > Compiling @openzeppelin/contracts-packages/protocol/ethereum-
package/contracts/utils/ReentrancyGuard.sol
 | > Compiling @openzeppelin/upgrades/contracts/Initializable.sol
 | > Compiling @sablier/shared-packages/protocol/contracts/compound/CarefulMath.sol
 | > Compiling @sablier/shared-packages/protocol/contracts/compound/EIP20Interface.sol
 | > Compiling @sablier/shared-packages/protocol/contracts/compound/Exponential.sol
 | > Compiling @sablier/shared-packages/protocol/contracts/interfaces/ICERC20.sol
 | > Compiling @sablier/shared-packages/protocol/contracts/lifecycle/OwnableWithoutRenounce.sol
 | > Compiling @sablier/shared-packages/protocol/contracts/lifecycle/PausableWithoutRenounce.sol
 | > Compiling @sablier/shared-packages/protocol/contracts/lifecycle/PauserRoleWithoutRenounce.sol
 | > Compiling @sablier/shared-packages/protocol/contracts/mocks/CERC20Mock.sol
 | > Compiling @sablier/shared-packages/protocol/contracts/mocks/ERC20Mock.sol
 | > Compiling @sablier/shared-packages/protocol/contracts/test/EvilERC20.sol
 | > Compiling @sablier/shared-packages/protocol/contracts/test/NonStandardERC20.sol
 |
 |
 |
 |   Contract: CTokenManager
 |     admin functions
 |       whitelistCToken
 |         when the caller is the admin
 |           when the cToken is not whitelisted
 |              ✓ whitelists the cToken (58ms)
 |              ✓ emits a whitelistctoken event (39ms)
 |           when the token is not a cToken
 |             ✓ reverts (45ms)
 |           when the cToken is whitelisted
 |             ✓ reverts (116ms)
 |         when the caller is not the admin
 |           ✓ reverts
 |       discardCToken
 |         when the caller is the admin
 |           when the cToken is whitelisted
 |              ✓ discards the cToken (47ms)
 |              ✓ emits a discardctoken event
 |           when the cToken is not whitelisted
 |             ✓ reverts (51ms)
 |         when the caller is not the admin
 |           ✓ reverts
 |     view functions
 |       isCToken
 |         when the cToken is whitelisted
 |           ✓ returns true
 |         when the cToken is not whitelisted
 |           ✓ returns false
 |
 |   Contract: Sablier
 |     initialization
 |        ✓ reverts when the cTokenManager contract is the zero address (45ms)
 |     admin functions
 |       updateFee
 |         when the caller is the admin
 |           when the fee is a valid percentage
 |              ✓ updates the fee (46ms)
 |           when the fee is not a valid percentage
 |             ✓ reverts
 |         when the caller is not the admin
 |           ✓ reverts
 |       takeEarnings
 |         when the caller is the admin
 |           when the cToken is whitelisted
 |             when the amount does not exceed the available balance
 |               when the amount is not zero
 |                 when the stream did start but not end
 |                    ✓ takes the earnings (274ms)
 |               when the amount is zero
 |                 ✓ reverts (51ms)
 |             when the amount exceeds the available balance
 |               ✓ reverts (59ms)
 |           when the cToken is not whitelisted
 |             ✓ reverts (47ms)
 |         when the caller is not the admin
 |           ✓ reverts
 |     view functions
 |       getStream
 |         when the stream does not exist
 |           ✓ reverts
 |       deltaOf
 |         when the stream exists
 |           when the stream did not start
 |             ✓ returns 0
 |           when the stream did start but not end
 |              ✓ returns the time the number of seconds that passed since the start time
 |           when the stream did end
 |              ✓ returns the difference between the stop time and the start time
 |         when the stream does not exist
 |           ✓ reverts
 |       balanceOf
 |         when the stream exists
 |           when the stream did not start
 |              ✓ returns the whole deposit for the sender of the stream (39ms)
 |              ✓ returns 0 for the recipient of the stream
 |              ✓ returns 0 for anyone else
 |           when the stream did start but not end
 |              ✓ returns the pro rata balance for the sender of the stream
 |              ✓ returns the pro rata balance for the recipient of the stream
 |              ✓ returns 0 for anyone else
 |           when the stream did end
 |              ✓ returns 0 for the sender of the stream
 |              ✓ returns the whole deposit for the recipient of the stream
 |              ✓ returns 0 for anyone else (44ms)
 |         when the stream does not exist
 |           ✓ reverts
 |       isCompoundingStream
 |         when the compounding stream exists
 |           ✓ returns true
 |         when the stream exists but is not compounding
 |           ✓ returns false
 |         when the stream does not exist
 |           ✓ reverts
 |       getCompoundingStream
 |         when the stream exists but is not compounding
 |           ✓ reverts
 |         when the stream does not exist
 |           ✓ reverts
 |       interestOf
 |         when the compounding stream does not exist
 |           ✓ returns 0
 |         when the stream does not exist
 |           ✓ reverts (46ms)
 |       getEarnings
 |         when the ctoken is not whitelisted
 |           ✓ reverts
 |     effects & interactions functions
 |       createStream
 |         when not paused
 |           when the recipient is valid
 |             when the token contract is erc20 compliant
 |               when the sablier contract has enough allowance
 |                 when the sender has enough tokens
 |                   when the deposit is valid
 |                     when the start time is after block.timestamp
 |                       when the stop time is after the start time
 |                         ✓ creates the stream (84ms)
 |                         ✓ transfers the tokens to the contract (93ms)
 |                         ✓ increases the stream next stream id (97ms)
 |                         ✓ emits a stream event (69ms)
 |                       when the stop time is not after the start time
 |                         ✓ reverts
 |                     when the start time is not after block.timestamp
 |                       ✓ reverts
 |                   when the deposit is not valid
 |                     when the deposit is zero
 |                       ✓ reverts (38ms)
 |                     when the deposit is smaller than the time delta
 |                       ✓ reverts (45ms)
 |                     when the deposit is not a multiple of the time delta
 |                       ✓ reverts (83ms)
 |                 when the sender does not have enough tokens
 |                   ✓ reverts (99ms)
 |               when the sablier contract does not have enough allowance
 |                 when the sender has enough tokens
 |                   ✓ reverts (128ms)
 |                 when the sender does not have enough tokens
 |                   ✓ reverts (138ms)
 |             when the token contract is not erc20
 |               when the token contract is non-compliant
 |                 ✓ reverts (82ms)
 |               when the token contract is the zero address
 |                 ✓ reverts (78ms)
 |           when the recipient is the caller itself
 |             ✓ reverts (44ms)
 |           when the recipient is the contract itself
 |             ✓ reverts
 |           when the recipient is the zero address
 |             ✓ reverts
 |         when paused
 |           ✓ reverts
 |       createCompoundingStream
 |         when not paused
 |           when the cToken is whitelisted
 |             when interest shares are valid
 |               ✓ creates the compounding stream (145ms)
 |               ✓ transfers the tokens to the contract (124ms)
 |               ✓ emits a createcompoundingstream event (104ms)
 |             when interest shares are not valid
 |               ✓ reverts (49ms)
 |           when the cToken is not whitelisted
 |             ✓ reverts (43ms)
 |         when paused
 |           ✓ reverts
 |       withdrawFromStream
 |         when the stream exists
 |           when the caller is the sender of the stream
 |             when not paused
 |               when the withdrawal amount is higher than 0
 |                 when the stream did not start
 |                   ✓ reverts (102ms)
 |                 when the stream did start but not end
 |                   when the withdrawal amount does not exceed the available balance
 |                     ✓ withdraws from the stream (104ms)
 |                     ✓ emits a withdrawfromstream event (77ms)
 |                     ✓ decreases the stream balance (137ms)
 |                   when the withdrawal amount exceeds the available balance
 |                     ✓ reverts (87ms)
 |                 when the stream did end
 |                   when the withdrawal amount does not exceed the available balance
 |                     when the balance is not withdrawn in full
 |                       ✓ withdraws from the stream (117ms)
 |                       ✓ emits a withdrawfromstream event (69ms)
 |                       ✓ decreases the stream balance (144ms)
 |                     when the balance is withdrawn in full
 |                       ✓ withdraws from the stream (110ms)
```

```
  |                        ✓ emits a withdrawfromstream event (119ms)
  |                        ✓ deletes the stream object (107ms)
  |                    when the withdrawal amount exceeds the available balance
  |                      ✓ reverts (86ms)
  |                when the withdrawal amount is zero
  |                  ✓ reverts (39ms)
  |              when paused
  |                ✓ reverts (53ms)
  |            when the caller is the recipient of the stream
  |              when not paused
  |                when the withdrawal amount is higher than 0
  |                  when the stream did not start
  |                    ✓ reverts (83ms)
  |                  when the stream did start but not end
  |                    when the withdrawal amount does not exceed the available balance
  |                      ✓ withdraws from the stream (107ms)
  |                      ✓ emits a withdrawfromstream event (89ms)
  |                      ✓ decreases the stream balance (152ms)
  |                    when the withdrawal amount exceeds the available balance
  |                      ✓ reverts (122ms)
  |                  when the stream did end
  |                    when the withdrawal amount does not exceed the available balance
  |                      when the balance is not withdrawn in full
  |                        ✓ withdraws from the stream (101ms)
  |                        ✓ emits a withdrawfromstream event (71ms)
  |                        ✓ decreases the stream balance (138ms)
  |                      when the balance is withdrawn in full
  |                        ✓ withdraws from the stream (107ms)
  |                        ✓ emits a withdrawfromstream event (81ms)
  |                        ✓ deletes the stream object (111ms)
  |                    when the withdrawal amount exceeds the available balance
  |                      ✓ reverts (87ms)
  |                when the withdrawal amount is zero
  |                  ✓ reverts (94ms)
  |              when paused
  |                ✓ reverts
  |            when the caller is not the sender or the recipient of the stream
  |              ✓ reverts (47ms)
  |          when the stream does not exist
  |            ✓ reverts (99ms)
  |        withdrawFromCompoundingStream
  |          when the sender's interest share is not zero and the recipient's interest share is not
zero
  |            when the sablier fee is not zero and is not 100
  |              when not paused
  |                when the stream did not start
  |                  ✓ reverts (90ms)
  |                when the stream did start but not end
  |                  ✓ withdraws from the stream (342ms)
  |                  ✓ pays the interest to the sender of the stream (305ms)
  |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(323ms)
  |                  ✓ pays the interest to the sablier contract (316ms)
  |                  ✓ emits a withdrawfromstream event (183ms)
  |                  ✓ emits a payinterest event (175ms)
  |                  ✓ decreases the stream balance (256ms)
  |                when the stream did end
  |                  ✓ withdraws from the stream (295ms)
  |                  ✓ pays the interest to the sender of the stream (362ms)
  |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(393ms)
  |                  ✓ pays the interest to the sablier contract (335ms)
  |                  ✓ emits a withdrawfromstream event (180ms)
  |                  ✓ emits a payinterest event (206ms)
  |                  ✓ deletes the stream objects (217ms)
  |              when paused
  |                ✓ reverts
  |            when the sablier fee is 0
  |              when not paused
  |                when the stream did not start
  |                  ✓ reverts (82ms)
  |                when the stream did start but not end
  |                  ✓ withdraws from the stream (242ms)
  |                  ✓ pays the interest to the sender of the stream (258ms)
  |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(257ms)
  |                  ✓ pays the interest to the sablier contract (297ms)
  |                  ✓ emits a withdrawfromstream event (199ms)
  |                  ✓ emits a payinterest event (149ms)
  |                  ✓ decreases the stream balance (229ms)
  |                when the stream did end
  |                  ✓ withdraws from the stream (271ms)
  |                  ✓ pays the interest to the sender of the stream (268ms)
  |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(291ms)
  |                  ✓ pays the interest to the sablier contract (317ms)
  |                  ✓ emits a withdrawfromstream event (235ms)
  |                  ✓ emits a payinterest event (173ms)
  |                  ✓ deletes the stream objects (214ms)
  |              when paused
  |                ✓ reverts
  |            when the sablier fee is 100
```

```
 |              when not paused
 |                when the stream did not start
 |                  ✓ reverts (91ms)
 |                when the stream did start but not end
 |                  ✓ withdraws from the stream (212ms)
 |                  ✓ pays the interest to the sender of the stream (209ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(224ms)
 |                  ✓ pays the interest to the sablier contract (281ms)
 |                  ✓ emits a withdrawfromstream event (124ms)
 |                  ✓ emits a payinterest event (127ms)
 |                  ✓ decreases the stream balance (195ms)
 |                when the stream did end
 |                  ✓ withdraws from the stream (239ms)
 |                  ✓ pays the interest to the sender of the stream (259ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(213ms)
 |                  ✓ pays the interest to the sablier contract (269ms)
 |                  ✓ emits a withdrawfromstream event (134ms)
 |                  ✓ emits a payinterest event (140ms)
 |                  ✓ deletes the stream objects (159ms)
 |              when paused
 |                ✓ reverts
 |          when the sender's interest share is zero
 |            when the sablier fee is not zero and is not 100
 |              when not paused
 |                when the stream did not start
 |                  ✓ reverts (83ms)
 |                when the stream did start but not end
 |                  ✓ withdraws from the stream (265ms)
 |                  ✓ pays the interest to the sender of the stream (272ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(282ms)
 |                  ✓ pays the interest to the sablier contract (306ms)
 |                  ✓ emits a withdrawfromstream event (144ms)
 |                  ✓ emits a payinterest event (182ms)
 |                  ✓ decreases the stream balance (211ms)
 |                when the stream did end
 |                  ✓ withdraws from the stream (261ms)
 |                  ✓ pays the interest to the sender of the stream (282ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(298ms)
 |                  ✓ pays the interest to the sablier contract (335ms)
 |                  ✓ emits a withdrawfromstream event (179ms)
 |                  ✓ emits a payinterest event (167ms)
 |                  ✓ deletes the stream objects (212ms)
 |              when paused
 |                ✓ reverts (39ms)
 |            when the sablier fee is 0
 |              when not paused
 |                when the stream did not start
 |                  ✓ reverts (85ms)
 |                when the stream did start but not end
 |                  ✓ withdraws from the stream (242ms)
 |                  ✓ pays the interest to the sender of the stream (260ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(277ms)
 |                  ✓ pays the interest to the sablier contract (311ms)
 |                  ✓ emits a withdrawfromstream event (156ms)
 |                  ✓ emits a payinterest event (146ms)
 |                  ✓ decreases the stream balance (216ms)
 |                when the stream did end
 |                  ✓ withdraws from the stream (250ms)
 |                  ✓ pays the interest to the sender of the stream (251ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(275ms)
 |                  ✓ pays the interest to the sablier contract (306ms)
 |                  ✓ emits a withdrawfromstream event (161ms)
 |                  ✓ emits a payinterest event (155ms)
 |                  ✓ deletes the stream objects (208ms)
 |              when paused
 |                ✓ reverts
 |            when the sablier fee is 100
 |              when not paused
 |                when the stream did not start
 |                  ✓ reverts (89ms)
 |                when the stream did start but not end
 |                  ✓ withdraws from the stream (222ms)
 |                  ✓ pays the interest to the sender of the stream (200ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(213ms)
 |                  ✓ pays the interest to the sablier contract (256ms)
 |                  ✓ emits a withdrawfromstream event (127ms)
 |                  ✓ emits a payinterest event (125ms)
 |                  ✓ decreases the stream balance (185ms)
 |                when the stream did end
 |                  ✓ withdraws from the stream (236ms)
 |                  ✓ pays the interest to the sender of the stream (286ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(224ms)
 |                  ✓ pays the interest to the sablier contract (270ms)
 |                  ✓ emits a withdrawfromstream event (132ms)
 |                  ✓ emits a payinterest event (136ms)
 |                  ✓ deletes the stream objects (160ms)
 |              when paused
 |                ✓ reverts
 |          when the recipient's interest share is zero
 |            when the sablier fee is not zero and is not 100
 |              when not paused
 |                when the stream did not start
 |                  ✓ reverts (83ms)
 |                when the stream did start but not end
 |                  ✓ withdraws from the stream (252ms)
 |                  ✓ pays the interest to the sender of the stream (285ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(264ms)
 |                  ✓ pays the interest to the sablier contract (329ms)
 |                  ✓ emits a withdrawfromstream event (156ms)
 |                  ✓ emits a payinterest event (163ms)
 |                  ✓ decreases the stream balance (233ms)
 |                when the stream did end
 |                  ✓ withdraws from the stream (256ms)
 |                  ✓ pays the interest to the sender of the stream (424ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(426ms)
 |                  ✓ pays the interest to the sablier contract (386ms)
 |                  ✓ emits a withdrawfromstream event (175ms)
 |                  ✓ emits a payinterest event (185ms)
 |                  ✓ deletes the stream objects (219ms)
 |              when paused
 |                ✓ reverts
 |            when the sablier fee is 0
 |              when not paused
 |                when the stream did not start
 |                  ✓ reverts (83ms)
 |                when the stream did start but not end
 |                  ✓ withdraws from the stream (259ms)
 |                  ✓ pays the interest to the sender of the stream (259ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(254ms)
 |                  ✓ pays the interest to the sablier contract (357ms)
 |                  ✓ emits a withdrawfromstream event (152ms)
 |                  ✓ emits a payinterest event (169ms)
 |                  ✓ decreases the stream balance (211ms)
 |                when the stream did end
 |                  ✓ withdraws from the stream (250ms)
 |                  ✓ pays the interest to the sender of the stream (272ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(267ms)
 |                  ✓ pays the interest to the sablier contract (303ms)
 |                  ✓ emits a withdrawfromstream event (164ms)
 |                  ✓ emits a payinterest event (177ms)
 |                  ✓ deletes the stream objects (201ms)
 |              when paused
 |                ✓ reverts
 |            when the sablier fee is 100
 |              when not paused
 |                when the stream did not start
 |                  ✓ reverts (82ms)
 |                when the stream did start but not end
 |                  ✓ withdraws from the stream (221ms)
 |                  ✓ pays the interest to the sender of the stream (199ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(208ms)
 |                  ✓ pays the interest to the sablier contract (243ms)
 |                  ✓ emits a withdrawfromstream event (121ms)
 |                  ✓ emits a payinterest event (130ms)
 |                  ✓ decreases the stream balance (211ms)
 |                when the stream did end
 |                  ✓ withdraws from the stream (244ms)
 |                  ✓ pays the interest to the sender of the stream (213ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(267ms)
 |                  ✓ pays the interest to the sablier contract (264ms)
 |                  ✓ emits a withdrawfromstream event (151ms)
 |                  ✓ emits a payinterest event (140ms)
 |                  ✓ deletes the stream objects (163ms)
 |              when paused
 |                ✓ reverts
 |        cancelStream
 |          when the stream exists
 |            when the caller is the sender of the stream
 |              when the stream did not start
 |                ✓ cancels the stream (124ms)
 |                ✓ transfers all tokens to the sender of the stream (122ms)
 |                ✓ emits a cancel event (87ms)
 |              when the stream did start but not end
 |                ✓ cancels the stream (129ms)
 |                ✓ transfers the tokens to the sender of the stream (127ms)
 |                ✓ transfers the tokens to the recipient of the stream (147ms)
 |                ✓ emits a cancel event (103ms)
 |              when the stream did end
 |                ✓ cancels the stream (108ms)
 |                ✓ transfers nothing to the sender of the stream (116ms)
 |                 ✓ transfers all tokens to the recipient of the stream (114ms)
 |                 ✓ emits a cancel event (89ms)
 |            when the caller is the recipient of the stream
 |              when the stream did not start
 |                 ✓ cancels the stream (109ms)
 |                 ✓ transfers all tokens to the sender of the stream (119ms)
 |                 ✓ emits a cancel event (99ms)
 |              when the stream did start but not end
 |                 ✓ cancels the stream (128ms)
 |                 ✓ transfers the tokens to the sender of the stream (131ms)
 |                 ✓ transfers the tokens to the recipient of the stream (127ms)
 |                 ✓ emits a cancel event (101ms)
 |              when the stream did end
 |                 ✓ cancels the stream (111ms)
 |                 ✓ transfers nothing to the sender of the stream (117ms)
 |                 ✓ transfers all tokens to the recipient of the stream (114ms)
 |                 ✓ emits a cancel event (91ms)
 |            when the caller is not the sender or the recipient of the stream
 |               ✓ reverts (40ms)
 |          when the stream does not exist
 |             ✓ reverts (46ms)
 |        cancelCompoundingStream
 |          when the sender's interest share is not zero and the recipient's interest share is not
zero
 |            when the sablier fee is not zero and is not 100
 |              when there were no withdrawals
 |                when the stream did not start
 |                   ✓ cancels the stream (215ms)
 |                when the stream did start but not end
 |                   ✓ cancels the stream (224ms)
 |                   ✓ transfers the tokens and pays the interest to the sender of the stream (462ms)
 |                   ✓ transfers the tokens and pays the interest to the recipient of the stream
(306ms)
 |                   ✓ pays the interest to the sablier contract (395ms)
 |                   ✓ emits a cancelstream event (200ms)
 |                   ✓ emits a payinterest event (186ms)
 |                when the stream did end
 |                   ✓ cancels the stream (224ms)
 |                   ✓ transfers the tokens and pays the interest to the sender of the stream (318ms)
 |                   ✓ transfers the tokens and pays the interest to the recipient of the stream
(315ms)
 |                   ✓ pays the interest to the sablier contract (362ms)
 |                   ✓ emits a cancelstream event (323ms)
 |                   ✓ emits a payinterest event (239ms)
 |              when there were withdrawals
 |                when the stream did start but not end
 |                   ✓ cancels the stream (216ms)
 |            when the sablier fee is 0
 |              when there were no withdrawals
 |                when the stream did not start
 |                   ✓ cancels the stream (240ms)
 |                when the stream did start but not end
 |                   ✓ cancels the stream (205ms)
 |                   ✓ transfers the tokens and pays the interest to the sender of the stream (282ms)
 |                   ✓ transfers the tokens and pays the interest to the recipient of the stream
(289ms)
 |                   ✓ pays the interest to the sablier contract (359ms)
 |                   ✓ emits a cancelstream event (198ms)
 |                   ✓ emits a payinterest event (177ms)
 |                when the stream did end
 |                   ✓ cancels the stream (219ms)
 |                   ✓ transfers the tokens and pays the interest to the sender of the stream (320ms)
 |                   ✓ transfers the tokens and pays the interest to the recipient of the stream
(288ms)
 |                   ✓ pays the interest to the sablier contract (397ms)
 |                   ✓ emits a cancelstream event (171ms)
 |                   ✓ emits a payinterest event (196ms)
 |              when there were withdrawals
 |                when the stream did start but not end
 |                   ✓ cancels the stream (203ms)
 |            when the sablier fee is 100
 |              when there were no withdrawals
 |                when the stream did not start
 |                   ✓ cancels the stream (176ms)
 |                when the stream did start but not end
 |                   ✓ cancels the stream (197ms)
 |                   ✓ transfers the tokens and pays the interest to the sender of the stream (254ms)
 |                   ✓ transfers the tokens and pays the interest to the recipient of the stream
(236ms)
 |                   ✓ pays the interest to the sablier contract (302ms)
 |                   ✓ emits a cancelstream event (169ms)
 |                   ✓ emits a payinterest event (155ms)
 |                when the stream did end
 |                   ✓ cancels the stream (195ms)
 |                   ✓ transfers the tokens and pays the interest to the sender of the stream (220ms)
 |                   ✓ transfers the tokens and pays the interest to the recipient of the stream
(241ms)
 |                   ✓ pays the interest to the sablier contract (341ms)
 |                   ✓ emits a cancelstream event (159ms)
 |                   ✓ emits a payinterest event (167ms)
 |              when there were withdrawals
 |                when the stream did start but not end
 |                   ✓ cancels the stream (203ms)
 |         when the sender's interest share is zero
 |           when the sablier fee is not zero and is not 100
 |             when there were no withdrawals
 |               when the stream did not start
 |                  ✓ cancels the stream (210ms)
 |               when the stream did start but not end
 |                  ✓ cancels the stream (218ms)
 |                  ✓ transfers the tokens and pays the interest to the sender of the stream (310ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(354ms)
 |                  ✓ pays the interest to the sablier contract (347ms)
 |                  ✓ emits a cancelstream event (189ms)
 |                  ✓ emits a payinterest event (185ms)
 |               when the stream did end
 |                  ✓ cancels the stream (214ms)
 |                  ✓ transfers the tokens and pays the interest to the sender of the stream (302ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(292ms)
 |                  ✓ pays the interest to the sablier contract (410ms)
 |                  ✓ emits a cancelstream event (175ms)
 |                  ✓ emits a payinterest event (215ms)
 |             when there were withdrawals
 |               when the stream did start but not end
 |                  ✓ cancels the stream (210ms)
 |           when the sablier fee is 0
 |             when there were no withdrawals
 |               when the stream did not start
 |                  ✓ cancels the stream (215ms)
 |               when the stream did start but not end
 |                  ✓ cancels the stream (200ms)
 |                  ✓ transfers the tokens and pays the interest to the sender of the stream (281ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(287ms)
 |                  ✓ pays the interest to the sablier contract (336ms)
 |                  ✓ emits a cancelstream event (231ms)
 |                  ✓ emits a payinterest event (168ms)
 |               when the stream did end
 |                  ✓ cancels the stream (236ms)
 |                  ✓ transfers the tokens and pays the interest to the sender of the stream (259ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(340ms)
 |                  ✓ pays the interest to the sablier contract (334ms)
 |                  ✓ emits a cancelstream event (160ms)
 |                  ✓ emits a payinterest event (154ms)
 |             when there were withdrawals
 |               when the stream did start but not end
 |                  ✓ cancels the stream (206ms)
 |           when the sablier fee is 100
 |             when there were no withdrawals
 |               when the stream did not start
 |                  ✓ cancels the stream (178ms)
 |               when the stream did start but not end
 |                  ✓ cancels the stream (183ms)
 |                  ✓ transfers the tokens and pays the interest to the sender of the stream (258ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(247ms)
 |                  ✓ pays the interest to the sablier contract (311ms)
 |                  ✓ emits a cancelstream event (159ms)
 |                  ✓ emits a payinterest event (165ms)
 |               when the stream did end
 |                  ✓ cancels the stream (179ms)
 |                  ✓ transfers the tokens and pays the interest to the sender of the stream (227ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(223ms)
 |                  ✓ pays the interest to the sablier contract (288ms)
 |                  ✓ emits a cancelstream event (198ms)
 |                  ✓ emits a payinterest event (145ms)
 |             when there were withdrawals
 |               when the stream did start but not end
 |                  ✓ cancels the stream (183ms)
 |         when the recipient's interest share is zero
 |           when the sablier fee is not zero and is not 100
 |             when there were no withdrawals
 |               when the stream did not start
 |                  ✓ cancels the stream (202ms)
 |               when the stream did start but not end
 |                  ✓ cancels the stream (218ms)
 |                  ✓ transfers the tokens and pays the interest to the sender of the stream (311ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(313ms)
 |                  ✓ pays the interest to the sablier contract (404ms)
 |                  ✓ emits a cancelstream event (189ms)
 |                  ✓ emits a payinterest event (191ms)
 |               when the stream did end
 |                  ✓ cancels the stream (226ms)
 |                  ✓ transfers the tokens and pays the interest to the sender of the stream (302ms)
 |                  ✓ transfers the tokens and pays the interest to the recipient of the stream
(339ms)
 |                  ✓ pays the interest to the sablier contract (350ms)
 |                  ✓ emits a cancelstream event (194ms)
 |                  ✓ emits a payinterest event (182ms)
 |              when there were withdrawals
 |                when the stream did start but not end
 |                   ✓ cancels the stream (216ms)
 |            when the sablier fee is 0
 |              when there were no withdrawals
 |                when the stream did not start
 |                   ✓ cancels the stream (192ms)
 |                when the stream did start but not end
 |                   ✓ cancels the stream (222ms)
 |                   ✓ transfers the tokens and pays the interest to the sender of the stream (281ms)
 |                   ✓ transfers the tokens and pays the interest to the recipient of the stream
(285ms)
 |                   ✓ pays the interest to the sablier contract (342ms)
 |                   ✓ emits a cancelstream event (176ms)
 |                   ✓ emits a payinterest event (176ms)
 |                when the stream did end
 |                   ✓ cancels the stream (206ms)
 |                   ✓ transfers the tokens and pays the interest to the sender of the stream (294ms)
 |                   ✓ transfers the tokens and pays the interest to the recipient of the stream
(275ms)
 |                   ✓ pays the interest to the sablier contract (342ms)
 |                   ✓ emits a cancelstream event (227ms)
 |                   ✓ emits a payinterest event (176ms)
 |              when there were withdrawals
 |                when the stream did start but not end
 |                   ✓ cancels the stream (212ms)
 |            when the sablier fee is 100
 |              when there were no withdrawals
 |                when the stream did not start
 |                   ✓ cancels the stream (178ms)
 |                when the stream did start but not end
 |                   ✓ cancels the stream (195ms)
 |                   ✓ transfers the tokens and pays the interest to the sender of the stream (228ms)
 |                   ✓ transfers the tokens and pays the interest to the recipient of the stream
(242ms)
 |                   ✓ pays the interest to the sablier contract (340ms)
 |                   ✓ emits a cancelstream event (155ms)
 |                   ✓ emits a payinterest event (168ms)
 |                when the stream did end
 |                   ✓ cancels the stream (177ms)
 |                   ✓ transfers the tokens and pays the interest to the sender of the stream (226ms)
 |                   ✓ transfers the tokens and pays the interest to the recipient of the stream
(235ms)
 |                   ✓ pays the interest to the sablier contract (297ms)
 |                   ✓ emits a cancelstream event (140ms)
 |                   ✓ emits a payinterest event (144ms)
 |              when there were withdrawals
 |                when the stream did start but not end
 |                   ✓ cancels the stream (190ms)
 |
 |
 |   391 passing (5m)
 |
@sablier/payroll
$ packages/payroll/scripts/test.sh
 | Starting our own ganache instance
 | Waiting for ganache to launch on port 8545...
 | Connection to localhost port 8545 [tcp/*] succeeded!
 | Ganache launched!
$ /Users/ezulkosk/audits/sab2/node_modules/.bin/truffle version
 | You can improve web3's peformance when running Node.js versions older than 10.5.0 by installing
the (deprecated) scrypt package in your project
 | You can improve web3's peformance when running Node.js versions older than 10.5.0 by installing
the (deprecated) scrypt package in your project
 | Truffle v5.0.35 (core: 5.0.35)
 | Solidity - 0.5.11 (solc-js)
 | Node v8.11.4
 | Web3.js v1.2.1
$ /Users/ezulkosk/audits/sab2/node_modules/.bin/truffle test
 | You can improve web3's peformance when running Node.js versions older than 10.5.0 by installing
the (deprecated) scrypt package in your project
 | You can improve web3's peformance when running Node.js versions older than 10.5.0 by installing
the (deprecated) scrypt package in your project
 |
 | Compiling your contracts...
 | ===========================
 | > Compiling packages/payroll/contracts/Migrations.sol
 | > Compiling packages/payroll/contracts/Payroll.sol
 | > Compiling packages/payroll/contracts/test/Imports.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-package/contracts/GSN/Context.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-package/contracts/GSN/GSNContext.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-
package/contracts/GSN/GSNRecipient.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-package/contracts/GSN/IRelayHub.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-
package/contracts/GSN/IRelayRecipient.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-
package/contracts/GSN/bouncers/GSNBouncerBase.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-
package/contracts/GSN/bouncers/GSNBouncerSignature.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-package/contracts/access/Roles.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-
```

```
package/contracts/access/roles/MinterRole.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-
package/contracts/cryptography/ECDSA.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-package/contracts/math/SafeMath.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-
package/contracts/token/ERC20/ERC20.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-
package/contracts/token/ERC20/ERC20Mintable.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-
package/contracts/token/ERC20/IERC20.sol
 | > Compiling @openzeppelin/contracts-packages/payroll/ethereum-
package/contracts/utils/ReentrancyGuard.sol
 | > Compiling @openzeppelin/upgrades/contracts/Initializable.sol
 | > Compiling @sablier/protocol/contracts/CTokenManager.sol
 | > Compiling @sablier/protocol/contracts/Sablier.sol
 | > Compiling @sablier/protocol/contracts/Types.sol
 | > Compiling @sablier/protocol/contracts/interfaces/ICTokenManager.sol
 | > Compiling @sablier/protocol/contracts/interfaces/IERC1620.sol
 | > Compiling @sablier/shared-packages/payroll/contracts/compound/CarefulMath.sol
 | > Compiling @sablier/shared-packages/payroll/contracts/compound/EIP20Interface.sol
 | > Compiling @sablier/shared-packages/payroll/contracts/compound/Exponential.sol
 | > Compiling @sablier/shared-packages/payroll/contracts/interfaces/ICERC20.sol
 | > Compiling @sablier/shared-packages/payroll/contracts/lifecycle/OwnableWithoutRenounce.sol
 | > Compiling @sablier/shared-packages/payroll/contracts/lifecycle/PausableWithoutRenounce.sol
 | > Compiling @sablier/shared-packages/payroll/contracts/lifecycle/PauserRoleWithoutRenounce.sol
 | > Compiling @sablier/shared-packages/payroll/contracts/mocks/CERC20Mock.sol
 | > Compiling @sablier/shared-packages/payroll/contracts/mocks/ERC20Mock.sol
 | > Compiling @sablier/shared-packages/payroll/contracts/test/EvilERC20.sol
 | > Compiling @sablier/shared-packages/payroll/contracts/test/NonStandardERC20.sol
 |
 | Please set the SABLIER_ADDRESS environment variable
 |
 |
 |   Contract: Payroll
 |     initialization
 |        ✓ reverts when the owner is the zero address (89ms)
 |        ✓ reverts when the signer is the zero address (80ms)
 |        ✓ reverts when the sablier contract is the zero address (123ms)
 |     admin functions
 |       whitelistRelayer
 |         when the salary exists
 |           when the relayer is not whitelisted
 |              ✓ whitelists the relayer (49ms)
 |           when the relayer is whitelisted
 |              ✓ reverts (69ms)
 |         when the salary does not exist
 |            ✓ reverts (38ms)
 |       discardRelayer
 |         discardRelayer
 |           when the salary exists
 |             when the relayer is whitelisted
 |                ✓ removes the relayer (75ms)
 |             when the relayer is not whitelisted
 |                ✓ reverts (39ms)
 |     view functions
 |       getSalary
 |         when the salary does not exist
 |            ✓ reverts
 |     effects & interactions functions
 |       createSalary
 |         when the token contract is erc20 compliant
 |           when the payroll contract has enough allowance
 |              ✓ creates the salary (136ms)
 |              ✓ transfers the tokens to the contract (133ms)
 |              ✓ increases the next salary id (137ms)
 |              ✓ emits a createsalary event (100ms)
 |           when the payroll contract does not have enough allowance
 |             when the company has enough tokens
 |                ✓ reverts (62ms)
 |             when the company does not have enough tokens
 |                ✓ reverts (52ms)
 |         when the token contract is not erc20
 |           when the token contract is non-compliant
 |              ✓ reverts (39ms)
 |           when the token contract is the zero address
 |              ✓ reverts
 |       createCompoundingSalary
 |         when the cToken is whitelisted
 |           when interest shares are valid
 |              ✓ creates the compounding salary (207ms)
 |              ✓ transfers the tokens to the contract (172ms)
 |              ✓ increases the next salary id (157ms)
 |              ✓ emits a createsalary event (131ms)
 |           when interest shares are not valid
 |              ✓ reverts (51ms)
 |         when the cToken is not whitelisted
 |            ✓ reverts (104ms)
 |       withdrawFromSalary
 |         when the salary exists
 |           when the caller is the employee
 |             when the stream did start but not end
 |               when the withdrawal amount is within the available balance
 |                  ✓ makes the withdrawal (120ms)
 |                  ✓ emits a withdrawfromsalary event (111ms)
 |                  ✓ decreases the stream balance (162ms)
 |               when the withdrawal amount is not within the available balance
 |                  ✓ reverts (127ms)
 |           when the caller is a relayer
 |             when the stream did start but not end
 |               when the withdrawal amount is within the available balance
 |                  ✓ makes the withdrawal (121ms)
 |                  ✓ emits a withdrawfromsalary event (94ms)
 |                  ✓ decreases the stream balance (162ms)
 |               when the withdrawal amount is not within the available balance
 |                  ✓ reverts (129ms)
 |           when the caller is not the employee or a relayer
 |              ✓ reverts (63ms)
 |         when the salary does not exist
 |            ✓ reverts
 |       cancelSalary
 |         when the salary exists
 |           when the caller is the company
 |             when the stream did start but not end
 |                ✓ cancels the salary (186ms)
 |                ✓ transfers the tokens to the sender of the stream (212ms)
 |                ✓ transfers the tokens to the recipient of the stream (197ms)
 |                ✓ emits a cancelsalary event (170ms)
 |           when the caller is the employee
 |             when the stream did start but not end
 |                ✓ cancels the salary (182ms)
 |                ✓ transfers the tokens to the sender of the stream (216ms)
 |                ✓ transfers the tokens to the recipient of the stream (196ms)
 |                ✓ emits a cancelsalary event (180ms)
 |           when the caller is not the company or the employee
 |              ✓ reverts (56ms)
 |         when the salary does not exist
 |            ✓ reverts
 |       cancelCompoundingSalary
 |         when the salary exists
 |           when the caller is the company
 |             when the stream did start but not end
 |                ✓ cancels the compounding salary (388ms)
 |                ✓ transfers the tokens and pays the interest to the company (433ms)
 |                ✓ transfers the tokens and pays the interest to the employee (452ms)
 |                ✓ pays the interest to the sablier contract (532ms)
 |                ✓ emits a cancelsalary event (334ms)
 |           when the caller is the employee
 |             when the stream did start but not end
 |                ✓ cancels the compounding salary (379ms)
 |                ✓ transfers the tokens and pays the interest to the company (438ms)
 |                ✓ transfers the tokens and pays the interest to the employee (490ms)
 |                ✓ pays the interest to the sablier contract (495ms)
 |                ✓ emits a cancelsalary event (326ms)
 |           when the caller is not the company or the employee
 |              ✓ reverts (61ms)
 |         when the salary does not exist
 |            ✓ reverts
 |
 |
 |   55 passing (44s)
 |
 ✓  Done in 402.13s.
```

Code Coverage

The test suite is of high quality and covers the majority of lines and branches in the contracts. We recommend adding additional tests to improve the coverage to as close to 100% as possible. In particular, the following parts could require further tests:

- The `Payroll.acceptRelayedCall()` function is not covered.
- Several require-statements are missing coverage for the else-case (particularly when a `MathError` occurs or a `transfer` fails), however these are very minor omissions in the coverage.

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
packages/protocol/contracts/	99.52	75.61	100	99.53	
CTokenManager.sol	100	100	100	100	
Sablier.sol	99.5	74.68	100	99.51	367
Types.sol	100	100	100	100	
All files	99.52	75.61	100	99.53	

File	% Stmts	% Branch	% Funcs	% Lines	Uncovered Lines
packages/payroll/contracts/	93.33	73.68	91.67	93.75	
Payroll.sol	93.33	73.68	91.67	93.75	190,201,202,204
All files	**93.33**	**73.68**	**91.67**	**93.75**	

Appendix

File Signatures

The following are the SHA-256 hashes of the audited contracts and/or test files. A smart contract or file with a different SHA-256 hash has been modified, intentionally or otherwise, after the audit. You are cautioned that a different SHA-256 hash could be (but is not necessarily) an indication of a changed condition or potential vulnerability that was not within the scope of the audit.

Contracts

```
ae94b6be680488c6c35e94c59e11e0d19f74598cb80f5257bb77eb73e1bc3435 ./contracts_upload/ICERC20.sol

bcf9b73d9105e3eab51b6896110f885ff5da6fe8418ad5c7dd40da949736a775 ./contracts_upload/Exponential.sol

f682e854c4e627aae6d39d77309c23d70b6f40c0133b3fe2d47c444487edac44 ./contracts_upload/Sablier.sol

7832d2338a03f79872c36cce6173d36e01d5fb618a6028e8aa22dc45a491c1ad
./contracts_upload/OwnableWithoutRenounce.sol

3e78fa8571f7b66121339a7b42cd80964eebb8a8d11c02fd82a93b4d7a320616 ./contracts_upload/Payroll.sol

91a3f268ba340a8bf72c7905f9ae77629e6b1354042d7557d3f06796726519eb ./contracts_upload/Migrations.sol

9f6f4278f18c0f1dbac497bdcba48b47ef5773be79f466b6fea632e08c82b5e0 ./contracts_upload/CarefulMath.sol

983cbb929dc9d9065ce0c98442c9d5e232c0a792d09316b9155f5afa8ad7b9f5
./contracts_upload/EIP20Interface.sol

2689cc50c47cfa4ed02e6fed1c58951135632b7a44d98f5d8dbd3248b95baa0d ./contracts_upload/CTokenManager.sol

d3209968dd1442a6ee2780c8b3ec7ce70daf2157c4216a5a1f306c103fefaa2e
./contracts_upload/PauserRoleWithoutRenounce.sol

b3247f76d202815f8ec96da60783996b8b63670f477c6e28dbd94a8b9c137736
./contracts_upload/PausableWithoutRenounce.sol

098626094a949eae7c56b17623879364bbc4db168b8fe1b7991fa8825b6645ea ./contracts_upload/Types.sol

74805515023d793453220eb72d1b513a695b4c99994ace3206db8d42d8b55e07
./contracts_upload/interfaces/IERC1620.sol

b5ea94fe5d50c974f13599e4ebc4b69e91b2bcd7f6d73ab75b2799de211dcee9
./contracts_upload/interfaces/ICTokenManager.sol

94118b8c4ef291c381769097dc07c7cef6f78e116c71f313981975d2a91337bf ./contracts_upload/test/Imports.sol
```

Tests

```
533a87afa4e452be8a28fd844e1535260ade0c84e07c43609955b57dcfc1a8a8 ./Sablier.behavior.js

dcaa088fdfa9a01ca7b965785ff80086b818096758826a664e663e8a251e2994 ./setup.js

49056d8e76a6f3cfee02dd0aa52ea778d12887c32a7cdc0724b724b26457683c ./Payroll.js

b64c71a9d008ddd9df4846b1b4c15a5eb6f83206a2acb74336c10ee2cfab09dc ./Payroll.behavior.js

d8bc1a3fa23b5ec1bd9bdbf0474cbfff7bb0adf96439fb35358fa1da02053031 ./tests_upload/setup.js

d8740fa5b8056ef80dee25e06a4a64989808caf2ebcc3f53d180b307b83a80f6
./tests_upload/payroll/effects/compoundingSalary/CancelCompoundingSalary.js

03e091ca95237382c7d1dfd3129db539bd89a41658e7ccaa74670cd15a770582
./tests_upload/payroll/effects/compoundingSalary/CreateCompoundingSalary.js

8a6322113fa318334f22c39f791def87b5c4a5cd09779232da49af591a8aa9d8
./tests_upload/payroll/effects/salary/CancelSalary.js

7c83e6e7622fe4b492b2f096bbd2d26c7d1b2a1d7b93f24767acaa054417185b
./tests_upload/payroll/effects/salary/WithdrawFromSalary.js

ecb58c6c47c68a8cbfd86412355be01a4d17abf2fda544f5f0f9ee27dcba9a56
./tests_upload/payroll/effects/salary/CreateSalary.js

2d924699379bd07834f7f204f376823cdbe4176440602dd7386e338f59d7ff89
./tests_upload/payroll/view/GetSalary.js

7cf09af5872650d1d5129141ae007cd5e7506f02c0752f5562a1f4ab954a8562
./tests_upload/payroll/admin/WhitelistRelayer.js

d71211bf7989c97d2503b333be8791ef587c8a957e3f2b1268d88c4962e4223a
./tests_upload/payroll/admin/DiscardRelayer.js

3000898f2a2634e43405fcb710fbbe3c36b64effcd80b6c295c06dbcb77ce251 ./tests_upload/sablier/Sablier.js

5bf0446cff11964024be7c608d6a85c3e616ce5076f66eb3362954056cd9d69b
./tests_upload/sablier/effects/compoundingStream/CreateCompoundingStream.js

50f9175e2d397c6aea50939e7c484e1b5bcbe453f9661b94efc05dcaed2a9572
./tests_upload/sablier/effects/compoundingStream/CancelCompoundingStream.js

bdc723b2263421cf425a4c7ad58ce1dac13c1cf9caca2315473a67b648f3df92
./tests_upload/sablier/effects/compoundingStream/WithdrawFromCompoundingStream.js

29296fd5db928eeeec9a70d0a79d3581f9a2bac8656f4cf1fd4edaf0c1e04309
./tests_upload/sablier/effects/stream/CreateStream.js

92fdb82fbeffa6f3da88fb6dba8342e880ebfd69a05bba50693eb94f5c192094
./tests_upload/sablier/effects/stream/WithdrawFromStream.js

515e587abdd65bcb95815a609ff5b000e26671a96b405be33a4566cb117a8e25
./tests_upload/sablier/effects/stream/CancelStream.js

bde93c8ec8a25465fc21cbc5cb9b314537cad3f4a520b4f4620e7ff010b48960
./tests_upload/sablier/view/DeltaOf.js

821a13c10a3df78b5773516ab16557835f9b23c69bbd49d397105d246fd16900
./tests_upload/sablier/view/GetCompoundingStream.js

23daa2606cd4be5152d4387efe6f67b7f9b0d433df9831e7321439704233d77f
./tests_upload/sablier/view/BalanceOf.js

ced3be004d7fbdd71aa8d8a97714ed9e8f9b905cc90ede24fdefe8a218a449b7
./tests_upload/sablier/view/GetStream.js

261900025b59c4fb2e398179d66c041bda4f3c75a7c345c6bffb553e7a0fa35a
./tests_upload/sablier/view/IsCompoundingStream.js

3253aaf1addb085c9f7eb05c34580dcec2f8a9f79a73269c7e16b5769e925bc4
./tests_upload/sablier/view/InterestOf.js

eb9ac3fce5d85d1f46daa10bc8d7656d2926fe80899d64f2d2b0233eaf7c1ffe
./tests_upload/sablier/view/GetEarnings.js

07ebb5b16f84695c3014e77ab43fda08261045a96812db8bf7bee8c289bf9ee6
./tests_upload/sablier/admin/UpdateFee.js

19f7cfcc216e85de1760a506873ef4e470a906afd699fb2165d14f8c2219ef1d
./tests_upload/sablier/admin/TakeEarnings.js

9e9dbeca3258b96b5b266fc05c16ad47e80e8c09dd7868a1913265400c583db0
./tests_upload/cTokenManager/CTokenManager.behavior.js

8e3e88c50b05042fdf274866e9870ba9207beb33fbaf4a10e2d9b4d29af1677b
./tests_upload/cTokenManager/CTokenManager.js
```

```
ef6b033ce771f3b6145dd949a56395f4e75f6837aed34e7a781cdae3210f03c1
./tests_upload/cTokenManager/view/IsCToken.js

6ff24f915bf8123445f8fe78b4b94e33e712e58c1457e145d2d97a65ed872331
./tests_upload/cTokenManager/admin/WhitelistCToken.js

eb1022c9821bba527985a0b3c29b1332cb6ae6e8732524f388cde79f6485f0b0
./tests_upload/cTokenManager/admin/DiscardCToken.js
```

About Quantstamp

Quantstamp is a Y Combinator-backed company that helps to secure smart contracts at scale using computer-aided reasoning tools, with a mission to help boost adoption of this exponentially growing technology.

Quantstamp's team boasts decades of combined experience in formal verification, static analysis, and software verification. Collectively, our individuals have over 500 Google scholar citations and numerous published papers. In its mission to proliferate development and adoption of blockchain applications, Quantstamp is also developing a new protocol for smart contract verification to help smart contract developers and projects worldwide to perform cost-effective smart contract security audits.

To date, Quantstamp has helped to secure hundreds of millions of dollars of transaction value in smart contracts and has assisted dozens of blockchain projects globally with its white glove security auditing services. As an evangelist of the blockchain ecosystem, Quantstamp assists core infrastructure projects and leading community initiatives such as the Ethereum Community Fund to expedite the adoption of blockchain technology.

Finally, Quantstamp's dedication to research and development in the form of collaborations with leading academic institutions such as National University of Singapore and MIT (Massachusetts Institute of Technology) reflects Quantstamp's commitment to enable world-class smart contract innovation.