Hake
Sablier Audit

Smart Contract & Operational Security Audit

21/03/2023 – 31/03/2023

Executive Summary

Findings:

Findings Severity	Count
High	0
Medium	2
Low	4
Gas	2

Audit Context:

Protocol Type	Token Streaming
Repo | Commit	https://github.com/sablier-labs/v2-core| 8bd57eb
SLOC	1368
Time Period	21 - 31/03/2023
Client Report Version	0.1

Qualitative Analysis:

Metric	Rating	Comments
Code complexity	**Excellent**	Sablier has not introduced significant code complexity, reducing likelihood of attack risks.
Documentation	**Excellent**	Sablier is very well documented.
Best practices	**Excellent**	Sablier makes great use of industry standards.
Centralization risks	**Average**	Sablier has some concerning privileges.
OpSec Practices	**Good**	Sablier is not at great risk from an OpSec perspective.

TABLE OF CONTENTS

Methodologies

All code in scope has been thoroughly manually reviewed and tested against multiple adversarial perspectives.

In addition, PRBMath functions in scope have been tested through differential fuzzing using Foundry and formal verification using Halmos wherever possible.

Findings

Medium Severity

[M-01] – Missing a second step when transferring admin role.

The current admin transfer process involves the current owner calling `transferAdmin()`. This function checks the new admin is not the zero address and proceeds to write the new admin's address into the `admin` state variable. If the nominated EOA account is not a valid account, it is entirely possible the current admin may accidentally transfer ownership to an uncontrolled account, breaking all functions with the `onlyAdmin()` modifier.

Adminable.sol – L33-43

```
function transferAdmin(address newAdmin⬆) public virtual override onlyAdmin {
    // Effects: update the admin.
    admin = newAdmin⬆;

    // Log the transfer of the admin.
    emit IAdminable.TransferAdmin({ oldAdmin: msg.sender, newAdmin⬆: newAdmin⬆ });
}
```

Recommended Mitigation

Implement a two-step process where the current admin nominates an account and the nominated account needs to call an acceptAdminTransfer() function for the transfer of admin to fully succeed. This ensures the nominated account is a valid and active address.

[M-02] – Centralization Risk

Admin can change `protocolFee` and `flashFee` at any time with no restrictions.

If `protocolFee` is accidentally set higher than `MAX_FEE` all new stream creations will be stopped, and user's relying on such creations will have to deal with the inconvenience. This is not ideal and enforcing the limit of `MAX_FEE` in the `setProtocolFee()` function might be a better option.

`flashFee()` has no enforced limit on it's value at any point, therefore the admin could front run a user and increase the fee to 100%, taking all the user's fund. This could lead to reputation damage.

Recommended Mitigation

Check fees don't exceed `MAX_FEE` when setting them in SablierV2Comptroller.sol `setProtocolFee()` and `setFlashFee()` functions.

QA / Low Severity

[L-01] – Using named returns and unnamed returns in the same function hurts readability and consistency.

6 instances:

SablierV2LockupLinear.sol#L162-L163

SablierV2LockupLinear.sol#L175-L177

SablierV2LockupLinear.sol#L195-L214

SablierV2LockupDynamic.sol#L172-L174

SablierV2LockupDynamic.sol#L186-L188

SablierV2LockupDynamic.sol#L206-L223

[L-02] – Declaring variables to default value is redundant.

Note: Doing so has no effect on gas.

4 instances:

SablierV2LockupDynamic.sol#L507-L508

SablierV2Lockup.sol#L237-L238

SablierV2Lockup.sol#L135-L136

Helpers.sol#L174-L175

[L-03] – `Require` or `revert` should be used instead of `assert`

According to solidity documentation:

“Assert should only be used to test for internal errors, and to check invariants. Properly functioning code should never create a Panic, not even on invalid external input. If this happens, then there is a bug in your contract which you should fix.”

5 instances:

SablierV2LockupDynamic.sol#L342

SablierV2LockupDynamic.sol#L367

SablierV2LockupDynamic.sol#L585

SablierV2LockupLinear.sol#L229

SablierV2LockupLinear.sol#L503

[L-04] – No time limit for stream durations.

`cliffs`, `startTime`, `endTime` and `segments` can be set without any boundaries.

In the case users make a mistake and for example, create an uncancellable stream for 10 years, instead of 1 year, there is no way to recover their funds.

Setting a limit of 1 year for example, could be a prudent measure to avoid potentially devastating circumstances for users.

Gas Improvements

[G-01] – Functions guaranteed to revert when called by normal users can be marked payable

If a function modifier such as `onlyAdmin()` is used, the function will revert if a normal user tries to pay the function. Marking the function as payable will lower the gas cost for legitimate callers because the compiler will not include checks for whether a payment was provided. The extra opcodes avoided are CALLVALUE(2), DUP1(3), ISZERO(3), PUSH2(3), JUMPI(10), PUSH1(3), DUP1(3), REVERT(0), JUMPDEST(1), POP(2), which costs an average of about **21 gas per call** to the function, in addition to the extra deployment cost.

6 instances:
SablierV2Lockup.sol#L169-L170

SablierV2Comptroller.sol#L64-L65

SablierV2Comptroller.sol#L74-L75

SablierV2Comptroller.sol#L89-L90

SablierV2Base.sol#L62-L63

SablierV2Base.sol#L80-L81

[G-02] – `x += y` costs more gas than `x = x + y` for state variables (-= too).

Using ` a = a + b` instead of `a += b` saves 113 gas. Also valid for subtractions.

7 instances:
SablierV2LockupLinear.sol#L353

SablierV2LockupLinear.sol#L427

SablierV2LockupLinear.sol#L495

SablierV2FlashLoan.sol#L161-L164

SablierV2LockupDynamic.sol#L423

SablierV2LockupDynamic.sol#L508

SablierV2LockupDynamic.sol#L577

[G-03] – Gas can be saved by re-ordering revert.

Place revert statement before calculating the `protocolFee` in order to save gas if `brokerFee` check reverts.

Helpers.sol#L36-L44

```
// Calculate the protocol fee amount.
// The cast to uint128 is safe because the maximum fee is hard-coded and it is always
amounts⬆.protocolFee = uint128(ud(totalAmount⬆).mul(protocolFee⬆).intoUint256());

// Checks: the broker fee is not greater than `maxFee`.
if (brokerFee⬆.gt(maxFee⬆)) {
    revert Errors.SablierV2Lockup_BrokerFeeTooHigh(brokerFee⬆, maxFee⬆);
}
```

Operational Security Practices Review

Feedback:

*Team answers have been omitted and can be found at the end of the document.

Because the team members are not anonymous, they can be physically targeted.

While frequenting co-working spaces:

- Don’t ever leave your computer unattended. Beware of malicious actors that can inject spy/malware via external ports.
- Never take your hardwallet or private key with you.
- Cover your keyboard when typing passwords. Camera feeds can be easily compromised, and people using wearable cameras could easily see what you have typed. Simply holding a hoodie over your keyboard should do the trick. You might look weird, but better safe than sorry. Remember that Sablier might one day hold millions if not billions of dollars. Stealing your password now could be an easy way to steal your funds later.

In case a team member gets hacked his account can be used to spread misinformation or malware to other team members. Stablishing unique codewords to uniquely verify their email as legitimate can be a great way to prevent further damage to the team.
To avoid getting hacked in the first place:

- Use a VPN (it hides location and identity): makes it hard for you to be targeted using a met- analysis of the platforms and websites you visit.
- Aliasing Emails: help you identify suspicious emails that shouldn't be received in a given account.
- Adblock (open source) can prevents malicious scam attacks when browsing.
- Disable auto media downloads in every app.
- Beware of Clippers: software that can replace an address in your clipboard to a very similar-looking hacker's address which has the same symbols in the beginning and in the end as your original address. So always double check.
- Do a password clearance every 2-6 months and renew all your passwords to fresh new unrelated passwords.

Criminals having the knowledge of your physical location and what assets are under your control can obviously end badly.
Example: https://twitter.com/mah_twittar/status/1444088022060896256
To prevent similar situations there a are few steps you can take:

- If you’re ordering pizza (or anything) with crypto, order it for pickup instead of delivery.
- When announcing your presence at an event via social media, keep in mind you are also announcing it to criminals that you will be there. So attend such events with a certain level of caution. Beware of possible kidnapping extraction points, invitations from strangers to attend nearby events and any suspicious or concerning situations/activities. This might sound like common sense, but deciding not to turn down that hall can make all the difference. If possible, always take a trusted person with you and periodically check each other’s where abouts. Events are supposed to be fun, so don’t let some extra caution get in the way of that.
- Don’t openly brag about how much money you are making.

Remember to never digitalize your keys in any way.
Always remember encrypted messaging apps can still be compromised if an attacker has physical access to your device.

On password managers:
Time and time again the internet has shown us that everything is hackable. Operating with that assumption I can only recommend using pen and paper or some other physical medium of managing your passwords. Although not practical is better than the alternative of being hacked.
Of course, this is not very realistic, as keeping multiple passwords in pen and paper is cumbersome, prone to error and a pain in the butt.
The next best alternative is using open source software. Bitwarden (I have no affiliation with them) is a well-regarded open source password manager that could be a great alternative to what you might be using at the moment (LastPass would be one to stay far away from).

Conclusion:

Sablier doesn't currently hold much sensitive information and doesn't have many funds at risk. If their security was compromised users and other team members wouldn't be heavily impacted. Overall Sablier has good OpSec practices in relation to their potential threats.

Feedback Summary:

- Be careful when frequenting co-working spaces and events.
- Don't use crypto accounts to get deliveries to your residence.
- Use the internet carefully to avoid getting hacked or being a victim to phishing-scams.
- Always lean towards using open source software to secure your information.

Some extra general tips:

- Never handle you wallets/keys when you are tired/stressed/sleepy.
- Always read the transaction you are signing.
- Avoid using jailbroken devices, porn websites (yeah I know) and unvetted browser plugins.

Team Questions and Answers:

What is the size of your team?
At the moment, the team is made up of five people.

Do they have a remote, in office or hybrid work arrangement?
We all work fully remote, with the occasional co-working space being used when members are in the same city.

Do any team members frequent remote co-working spaces?
Yes, frequency depends on each member.

Are they full-time, part-time, contractors or a combination?
Full-time.

Is one or more team members anonymous?
No.

What channels do you use to communicate and for what reason?
Slack (internal) and Signal (private). For business engagements we use Email, Discord and Telegram.

If email is one of them list what emails: e.g google, outlook, protonmail…
Google and Protonmail.

Is remaining anonymous a priority or a concern for the company or it's team members?
No.

Does one or more team members have access to private keys that can be potentially used for exploits?
There's no single team member with full access to our system. We use multisigs and try to limit the admin powers any single entity can have over the protocol itself.

What is at stake if the private keys get compromised? How many funds could be at risk if an admin key gets compromise?
In order to obtain any power over the protocol, one would have to compromise multiple key holders. At the moment the multisig is 2/3, but we may increase the number of signatories in the future, with signers located in different geographical regions.
Regarding funds, the protocol admin does not have any escape hatch to the funds locked up in the Sablier V2 (for more on this, see the Governance section in the docs). In the unlikely event of an admin takeover, only new streams/actions would be affected in the sense that the admin may increase the fee to the contract's hardcoded "MAX_FEE" (which will be 10%), and withdraw those exorbitant fees to their wallet.

Is it known to the public, at least to some extent who is in control of the private keys?
We won't make any public statement about who exactly on the team control the keys, but it should be pretty obvious that the initial Sablier founders (Gabriel and Paul) will have partial control over the protocol multisig.

Is privacy and anonymity a concern for team members? If yes, to what extent?
Naturally, privacy may be a concern up to a point (we are public individuals at the end of the day but we may like to keep personal things like location or business-related on-chain activity a bit more private). Anonymity is not a major concern.

How is this private key stored? Hot(software) wallet, Cold(hardware) wallet, physically(in what medium, like paper, metal or wood)?

Cold wallets paired with physical key storage (paper and metal).

Does the team hold sensitive information?
Apart from business strategy and/or internal activity the team does not deal with particularly sensitive information.

Does the team exchange sensitive information? If so through what channel?
Slack (business), Signal or other encrypted messaging apps for more sensitive data.

How much would it affect the business or team members if information got leaked?
No significant effects; the worst that would happen is that our competitors would find out what we're up to.

Do you follow any OpSec practices? If so which? (E.g Using hardware wallets or multisigs)
Yes: hardware wallets, 2FA everywhere, multisigs, personal request on Slack when submitting multisig transactions, using password managers.

Are you aware of SIMSwaps? If so, have you called your mobile phone carrier to prevent it?
We acknowledge the risks associated with SIM swaps and have implemented preventative measures to mitigate them. However, we tend to not use phone numbers for 2FA (when we can use a 2FA app, we use that over phone number verification).