Source Code Review of HTLC for SIBEX AG

Final Report and Management Summary

2019-12-12

CONFIDENTIAL

X41 D-SEC GmbH
Dennewartstr. 25-27
D-52068 Aachen
Amtsgericht Aachen: HRB19989

https://x41-dsec.de/
info@x41-dsec.de

Revision	*Date*	*Change*	*Author(s)*
1	2019-10-14	Initial Report	L. Gommans
2	2019-10-23	Findings	L. Gommans, L. Merino
3	2019-10-28	Finalization	L. Gommans, L. Merino
4	2019-12-10	Retest	L. Merino
5	2019-12-12	Further advice	L. Merino

Contents

Dashboard

Target	
Customer	SIBEX AG
Name	SIBEX HTLC Code
Type	Blockchain Technology
Version	Commit b4b9c150
Engagement	
Type	Source Code Review
Consultants	2: Luis Merino and Luc Gommans
Engagement Effort	12 days, 2019-10-14 to 2019-10-25
Total issues found	2

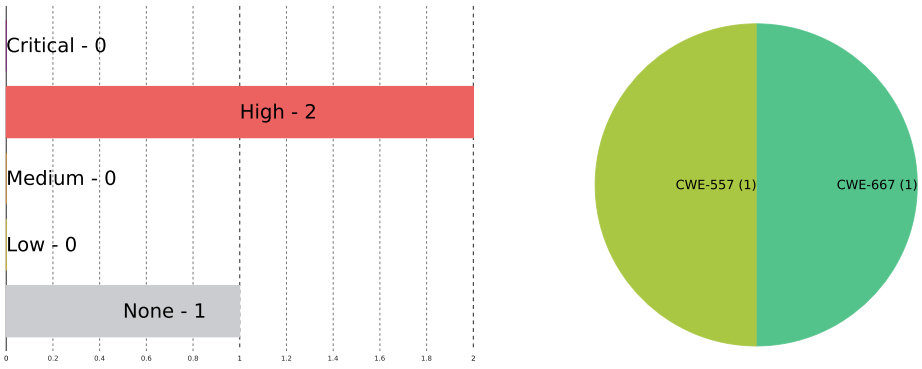

Figure 1: Issue Overview (l: Severity, r: CWE Distribution)

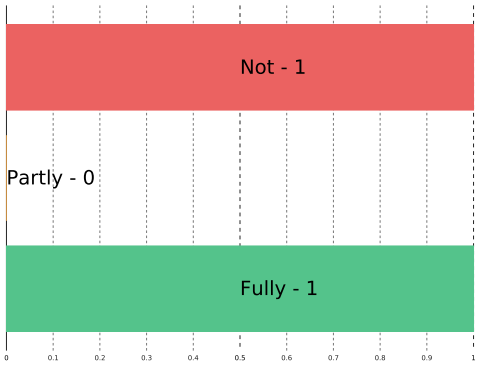

Figure 2: Remediation Results

1 Remediation Summary

In December 2019, X41 retested SIBEX HTLC Code in order to review the patches provided by SIBEX AG for the findings reported in an earlier stage.

X41 received a patch for Finding 5.1.1 that fully mitigates the problem, removing the trusted third party from running contracts in Ethereum and eliminating the risk of monetary loss by abusing the fees.

Regarding Finding 5.1.2, no technical solution is feasible at this time due to the limitations of Bitcoin Script. If the user is unable to act in time, monetary loss could occur. To mitigate the impact, X41 advises SIBEX AG to offer a Watchtower service which monitors all running contracts and send emergency alerts to the parties (via SMS, email, phone or other means) when they are at risk of monetary loss. Further advice is provided in order to mitigate the impact and reduce the risk. SIBEX AG disputes the rating and mitigation status of this finding. Given this software is aimed at professional traders and is expected to be deployed in high-availability cloud environments, they consider the risk is medium. Given there is no fix at reach that fully eliminates the risk, they consider the mitigation status is not-applicable.

X41 did not evaluate business risks or external factors in this Source Code Review.

2 Executive Summary

In October 2019, X41 D-Sec GmbH performed a Source Code Review against SIBEX HTLC Code of SIBEX AG to identify vulnerabilities and weaknesses in the design and implementation of their platform for Cross Chain Atomic Swaps.

A total of two vulnerabilities were discovered during the test by X41. None were rated as critical, two were classified as high severity, none as medium, and none as low. Additionally, one issue without a direct security impact was identified.

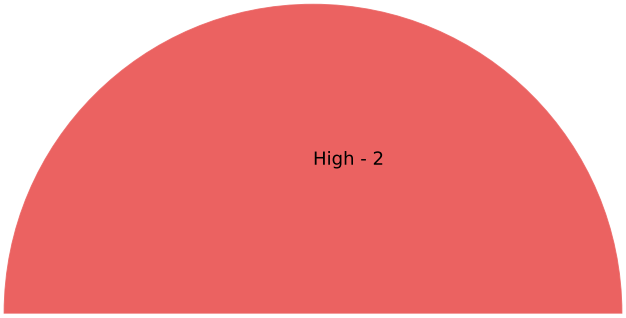

Figure 2.1: Issues and Severity

SIBEX AG provides a platform for Cross Chain Atomic Swaps, allowing customers to exchange cryptocurrencies with no third parties involved. Any serious vulnerabilities in the contracts could result in loss of funds to customers or SIBEX AG and have a negative impact on the reputation of SIBEX AG.

In a Source Code Review, the testers receive all available information about the project, including

source code. The test was performed by two experienced security experts between 2019-10-14 and 2019-10-25.

The issues identified allow an attacker to gain control of deposited coins in certain situations. One issue describes a flaw in the Solidity implementation, where an attacker controlling the fee address can steal all the customers deposits when contracts are redeemed. Another issue describes a flaw in the Bitcoin contract implementation, where contracts can be both redeemed and refunded after the timelock expires, leaving users in the Bitcoin chain in a handicapped position against their counterparties in the Ethereum chain, and allowing an attacker to steal both Bitcoin and Ether deposits when the participant does delay refunding an expired contract.

On a positive note cross chain transactions might be strengthened by HTLC, as it removes the trusted third party involved in exchange platforms. It also provides a solid framework guaranteeing a secure transaction. Nevertheless, some strict timing conditions must be followed that might result in a risk of loss when the involved parties don't act in time.

In conclusion, the framework can live up to the expectations of professional and institutional traders, which will put in place redundant and highly available systems to ensure no money loss is caused because of offline or degraded systems not acting in time.

3 Introduction

X41 reviewed the components of Sibex which are involved in Hashed Time-Locked Contracts (HTLC[1]) for Cross Chain Atomic Swaps (XCAT[2]) involving Bitcoin and Solidity code. This review focused on design and implementation flaws that could have a real world security impact.

These contracts are considered sensitive because they control how and when coins are exchanged during atomic swaps. A vulnerability in their design or implementation could lead to direct money and reputation loss.

Attackers could try to attack the atomic swaps by abusing weaknesses in the contracts deployed. If the contract design is not sound or the implementation is flawed, the coins taking part be redeemed in unexpected ways, either by any of the involved parties or external actors. Furthermore, any of the participants or external parties could have a privileged position that could be abused to force a money loss on the other party.

3.1 Methodology

X41 reviewed the HTLC contracts implemented in Solidity for Ethereum and Bitcoin Scripts for Bitcoin, together with the CLI[3] tools written in Go.

A manual approach for penetration testing and for code review is used by X41. This process is supported by tools such as static code analyzers. X41 adheres to established standards for source code reviewing and penetration testing. These are in particular the *CERT Secure Coding*[4] standards and the *Study - A Penetration Testing Model*[5] of the German Federal Office for Information Security.

[1] Hash Time Locked Contracts
[2] Cross Chain Atomic Swap
[3] Command line interpreter
[4] `https://wiki.sei.cmu.edu/confluence/display/seccode/SEI+CERT+Coding+Standards`
[5] `https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/Studies/Penetration/penetration_pdf.pdf?__blob=publicationFile&v=1`

3.2 Findings Overview

DESCRIPTION	REMEDIATED	SEVERITY	ID	REF
Owner can Steal Ethereum Deposits via Contract Fees	FULLY	HIGH	SIBEX-CR-19-01	5.1.1
Bitcoin Contracts can Still be Redeemed After Expiration	NOT	HIGH	SIBEX-CR-19-02	5.1.2
Bitcoin Redemption Fee Charged Client-Side	N/A	NONE	SIBEX-CR-19-100	5.2.1

Table 3.1: Security Relevant Findings

The 'remediated' column includes the results of the retest.

3.3 Scope

In the quote *20190801-00-00*, the scope was defined as commit *7dd076b8* at `https://gitlab.com/sibex/sibex-client/tree/master/htlc`. At the start of the review on 2019-10-14, X41 noticed there had been more than 700 new commits to the project since this commit. It was agreed with SIBEX AG that the latest commit will be used as scope of the test, which at the time was commit *b4b9c150* on the *develop* branch[6].

Additionally, SIBEX AG indicated that the following files are most relevant for the review:

- *btc-htlc.go*
- *erc20-htlc.go*
- *erc20-htlc.sol*
- *eth-htlc.go*
- *eth-htlc.sol*
- *htlc.go*

The files reviewed contained 3165 lines of Go code, and 251 lines of Solidity.

3.4 Recommended Further Tests

It is recommended to review all parts of the system involved. In the recent past, attacks have been observed against infrastructure and also development systems in the domain of crypto currencies

[6] `https://gitlab.com/sibex/sibex/tree/b4b9c150711013cfb6b6569271a39f6741a28e1b/htlc`

and trading. Therefore, X41 recommends to also review such systems to assure supply chain security.

4 Rating Methodology for Security Vulnerabilities

Security vulnerabilities are given a purely technical rating by the testers as they are discovered during the test. Business factors and financial risks for SIBEX AG are beyond the scope of a penetration test which focuses entirely on technical factors. Yet technical results from a penetration test may be an integral part of a general risk assessment. A penetration test is based on a limited time frame and only covers vulnerabilities and security issues which have been found in the given time, there is no claim for full coverage.

In total, five different ratings exist, which are as follows:

Severity Rating
None
Low
Medium
High
Critical

4.1 Common Weakness Enumeration

The CWE[1] is a set of software weaknesses that allows the categorization of vulnerabilities and weaknesses in software. If applicable, X41 provides the CWE-ID for each vulnerability that is discovered during a test.

[1] Common Weakness Enumeration

CWE is a very powerful method to categorize a vulnerability and to give general descriptions and solution advice on recurring vulnerability types. CWE is developed by *MITRE*[2]. More information can be found on the CWE website at `https://cwe.mitre.org/`.

[2] `https://www.mitre.org`

5 Results

This chapter describes results of this test, the security relevant findings are documented in Section 5.1. Additionally, findings without a direct security impact are documented in Section 5.2.

5.1 Findings

The following subsections describe findings with a direct security impact that were discovered during the test.

5.1.1 SIBEX-CR-19-01: Owner can Steal Ethereum Deposits via Contract Fees

Severity:	HIGH
Remediated:	FULLY
CWE:	*667 – Improper Locking*

5.1.1.1 Description

When a HTLC contract in Ethereum is redeemed, the contract `owner` (see `owner` in *erc20-htlc.sol* and *eth-htlc.sol*) receives a proportional fee. The fee is calculated during the contract redemption process via ***redeem()*** by dividing the contract `amount` by a global divisor `feeDiv` and the resulting quantity is sent to `feeAddress`, while the remaining coins are sent to the contract `receiver`.

The function ***setFeeDiv()*** allows setting a new value for `feeDiv` when invoked from the `owner` address. This new `feeDiv` divisor will globally affect all ongoing and newly created contracts.

A malicious actor controlling the `owner` account can abuse ***setFeeDiv()*** to steal all Ether from contracts redeemed after the fee change. Setting `feeDiv` to 1 will make all funds deposited in the contract go to `feeAddress` on ***redeem()***, while no funds will be sent to the `receiver` account. Furthermore, even when the `receiver` notices that the fee has an unfair value, they need the counterparty cooperation to recover their funds. When this situation applies, the only winning move is waiting for the contract expiration and ***refund()***.

By the stated facts, we believe that the `owner` (i.e. Sibex) holds a privileged position and must be a trusted party. At any point, if the `owner` becomes malicious, deposited funds are in risk.

5.1.1.2 Solution Advice

X41 advises locking the fee value on contract creation, so it cant́ be modified for running contracts. A feasible way of implementing this change would be adding the fee value to the `LockContract` structure when ***newHTLC()*** is called. Furthermore, CLI tools should show the assigned fee for a contract, so both parties can audit itś value and decide if they want to participate in the scheme, refund or not funding it at all. By implementing this change, the `owner` does not hold the privileged position anymore and cant́ hijack the funds of running contracts.

5.1.1.3 Retest Status

X41 had access to the fixes proposed to remediate this finding (see commits *bf81cd60*, *130920bf* and *5697cfd2*). These patches modify the contract to include the fee divisor as part of the constructor and stores it in the contract data structure to use it for the fee calculation during contract redemption. In case the fee divisor provided by the user does not match the one set by the `owner`, the contract construction fails.

Furthermore, the CLI tools have been modified to retrieve the fee divisor set by the `owner` and log its value to the console.

X41 consider this finding has been fully remediated with the proposed patches.

5.1.2 SIBEX-CR-19-02: Bitcoin Contracts can Still be Redeemed After Expiration

Severity:	HIGH
Remediated:	NOT
CWE:	*557 – Concurrency Issues*

5.1.2.1 Description

The current implementation of HTLC for Bitcoin (see ***atomicSwapContract()*** in *btc-htlc.go*) allows redeeming a contract independently if the timelock has expired or not. By contrast, the Ethereum implementation does only allow refunds after expiration.

When an atomic exchange between Ether and Bitcoin is initiated in the Ethereum chain, the initiator holds a privileged position versus the Bitcoin party. A malicious actor could initiate a contract in Ethereum and, after observing the Bitcoin funds have been sent, take no action until the Ethereum timelock expires. If the Bitcoin party has not refunded by then, the initiator can both redeem the Bitcoin and refund the Ether at the same time, taking all the coins with them and leaving the Bitcoin party with total loss. The same situation does not hold when the contract is initiated in the Bitcoin chain, as Ether can only be refunded and not redeemed after timelock expiration.

It is worth noting that, according to comments in *htlc/htlc.go*, the 3h time window before the initiator funds time lock expires is already *"a very crucial time period"* for a different reason: *"after the initiator checks that there are funds from the participant, he redeems it if possible. [...] The participant now has 3 h to redeem, if he does not redeem, he the initiator gets both ETH and BTC."*.

5.1.2.2 Solution Advice

Disallowing contract redemption in Bitcoin after the timelock has expired would be the technical solution to this problem, but this can't be implemented in Bitcoin Script, as it requires an opcode to check if a timelock has not expired without invalidating the whole operation. We understand such modifications in the core of Bitcoin fall outside of SIBEX AG scope.

Nevertheless, X41 recommends to properly communicate this time critical period to the users so they can react in time to avoid loss of funds. Making sure they refund their contracts as soon as they have expired when they have not been redeemed will effectively mitigate the impact of this finding.

5.1.2.3 Retest Status

This cannot be fixed on a technical level by SIBEX AG due to limitations in Bitcoin Script, thus the technical remediation level is 'not remediated'. If the user of the software is unable to act on time, monetary loss could occur. The organizational workaround to communicate the critical time period to customers as described above still holds.

X41 advises deploying SIBEX AG's software in high-availability topologies to make sure the nodes are connected and can react in time during the critical time periods and rendering the impact of either accidental or targeted attacks as low as possible. Furthermore, having a contingency plan that allows quick reaction during unexpected events while there are running contracts will help reducing the risk to minimum and acceptable levels.

It is worth noting that attackers might put an unexpected amount of effort on keeping users offline during ongoing transactions when enough money is at stake. There are documented long lasting DDoS attacks that caused severed congestion and outages over several days involving compromised low-cost consumer devices under the control of averagely skilled attackers[1,2].

To mitigate the impact during an attack or unexpected outage, X41 also advises SIBEX AG to offer a Watchtower service which monitors all running contracts and send emergency alerts to the parties (via SMS, email, phone or other means) when they are at risk of monetary loss.

SIBEX AG disputes the rating and mitigation status of this finding. Given this software is aimed at professional traders and is expected to be deployed in high-availability cloud environments, they consider the severity is medium. Given that there is currently no fix possible that fully eliminates the risk, they consider the remediation status to be 'not applicable'.

X41 did not evaluate business risks or external factors in this Source Code Review.

[1] https://www.ovh.com/world/news/articles/a2367.the-ddos-that-didnt-break-the-camels-vac
[2] https://protonmail.com/blog/a-brief-update-regarding-ongoing-ddos-incidents/

5.2 Side Findings

The following observations do not have a direct security impact, but are related to security hardening or affect functionality and other topics that are not directly related to security.

5.2.1 SIBEX-CR-19-100: Bitcoin Redemption Fee Charged Client-Side

5.2.1.1 Description

In the function ***RedeemHTLCFunds*** in *btc-htlc.go*, a fee is calculated and paid to SIBEX AG, as shown in listing 5.1. Unlike in Ethereum, there is no concept of a standalone smart contract that is under SIBEX AG's control. Instead, the instructions for the HTLC to work are attached to individual transactions. Since it is a strictly two-party system, the program must be run by the trader and the ***RedeemHTLCFunds*** function can therefore be modified.

With little effort, Bitcoin users may not pay any transaction fees to SIBEX AG.

```
1     sibexFee := amountRedeem / core.BtcHtlcSibexfeediv
2     amountMinusFee := amountRedeem - sibexFee
3     redeemTx.TxOut[0].Value = amountMinusFee
4     redeemTx.TxOut[1].Value = sibexFee
```

Listing 5.1: Code Used to Calculate and Pay Redemption Fee

5.2.1.2 Solution Advice

X41 is not aware of a solution to this problem that avoids making SIBEX AG a trusted third party or does not rely on security through obscurity.

6 About X41 D-Sec GmbH

X41 D-Sec GmbH is an expert provider for application security and penetration testing services. Having extensive industry experience and expertise in the area of information security, a strong core security team of world-class security experts enables X41 D-Sec GmbH to perform premium security services.

X41 has the following references that show their experience in the field:

- Review of the Mozilla Firefox updater[1]
- X41 Browser Security White Paper[2]
- Review of Cryptographic Protocols (Wire)[3]
- Identification of flaws in Fax Machines[4,5]
- SmartCard Stack Fuzzing[6]

The testers at X41 have extensive experience with penetration testing and red teaming exercises in complex environments. This includes enterprise environments with thousands of users and vendor infrastructures such as the Mozilla Firefox Updater (Balrog).

Fields of expertise in the area of application security encompass security-centered code reviews, binary reverse-engineering and vulnerability-discovery. Custom research and IT security consulting, as well as support services, are the core competencies of X41. The team has a strong technical background and performs security reviews of complex and high-profile applications such as Google Chrome and Microsoft Edge web browsers.

X41 D-Sec GmbH can be reached via `https://x41-dsec.de` or `mailto:info@x41-dsec.de`.

[1] `https://blog.mozilla.org/security/2018/10/09/trusting-the-delivery-of-firefox-updates/`
[2] `https://browser-security.x41-dsec.de/X41-Browser-Security-White-Paper.pdf`
[3] `https://www.x41-dsec.de/reports/Kudelski-X41-Wire-Report-phase1-20170208.pdf`
[4] `https://www.x41-dsec.de/lab/blog/fax/`
[5] `https://2018.zeronights.ru/en/reports/zero-fax-given/`
[6] `https://www.x41-dsec.de/lab/blog/smartcards/`

Acronyms

CLI Command line interpreter 7
CWE Common Weakness Enumeration 10
HTLC Hash Time Locked Contracts 7
XCAT Cross Chain Atomic Swap 7