Fine penetration tests for fine websites

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Audit-Report AERUM Smart Contracts 09.-10.2018

Cure53, Dr.-Ing. M. Heiderich, BSc. F. Fäßler

Index

Introduction

> *“AERUM is an infrastructure project aimed to develop a practical Blockchain 3.0 compatible with Ethereum - a high-performance scalable decentralized platform for B2B and B2C applications.”*
>
> From https://aerum.com/

This report documents the findings of a security assessment targeting the AERUM smart contracts and carried out by Cure53. The project comprised two rounds of auditing, with the first round completed in late September 2018 and the second taking place in mid-October of the same year. Observations and findings accumulated over the course of the entire two-rounds Cure53 project are included in this report, yet the primary focus has been placed on round two.

In terms of the coverage and scope, the first round focused on the basic *AerumToken* smart contract with minimal modification of *OpenZeppelin ERC20.* For this part of the assessment, no noteworthy findings were spotted and the documentation reflecting this has been sent to the AERUM team. In spite of no relevant discoveries, brief notes on the

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

proceeding and reports from round one are also included here for the sake of completeness. For the crucial second round of the project, the sources of the *AerumCrowdsale* contract were provided to Cure53 for audit. Alongside the contracts, AERUM also furnished Cure53 with information about the token and the *crowdsale*, mostly relying on the AERUM "*Litepaper*" and the project's website for this task.

The core audits of the AERUM contracts honed in on the typical security issues, programming pitfalls, logical bugs and other loopholes that could potentially put AERUM users at risk. Additionally, the public claims made by AERUM were compared to the technical side of the project, meaning the items and handling actually implemented in the contracts. It needs to be mentioned that two senior members of the Cure53 investigated the project's scope for 2.5 days.

Once confirmed on the scope, the findings were live-reported to the AERUM team and then discussed in an online meeting. After that, a document with additional notes from the in-house team at AERUM was shared with Cure53. These notes have been added to this report to provide a more complete picture of the project's goals and current state. Given the objectives of the AERUM compound, Cure53 believed it relevant to include various details in this report. To give a sense of the general range of findings, it can be noted that seven security-relevant discoveries were made on the scope, with vulnerabilities ranging from "*Low*" to "*High*" as regards their severities.

In the following sections, the report first sheds light on the scope by linking the repositories hosting the audited code. Next, the report discusses all tickets, incorporating the aforementioned amendments and notes. Each finding is discussed in considerable depth to facilitate next steps for the AERUM team. Finally, in light of the discoveries and exchanges, Cure53 issues a broader verdict and testifies to the impressions about the general security posture of the AERUM smart contract projects in concluding paragraphs.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Scope

- **AREUM Smart Contracts**
  - **Round 1:**
    - https://github.com/AERUMTechnology/governance/blob/master/contracts/token/AerumToken.sol
  - **Round 2:**
    - https://github.com/AERUMTechnology/crowdsale
    - Website with background info: https://aerum.com/
    - Light-Paper with additional background info: https://aerum.com/lite-paper

Round 1 Test Results

In the first round of the security audit, the AERUM Technology team provided Cure53 with Solidity code of the *AerumToken* contract. This contract has been built upon the OpenZeppelin *Ownable* and *PausableToken* contracts. In the initially shared sources the precise OpenZeppelin version was not defined, yet the AERUM team provided these details promptly.

Along the contracts' code, Cure53 also received documentation to verify that the token indeed implements what the AERUM Technology compound promises. This includes **ensuring that the token is pausable, not mintable, not burnable and the initial supply is set correctly.** Beyond verifying the claims, Cure53 also audited the technical security of the contracts. This means checking for typical ERC20 issues and general Solidity pitfalls. Moreover, AERUM Technology extended the base OpenZeppelin, so that it had to be ascertained that the alterations do not cause any security-relevant side effects.

The result of the first round of the security audit covering the *AerumToken* should be seen as very positive. Cure53 has verified that the *AerumToken* contract factually implements what AERUM Technology claims. All functions are pausable, while the tokens are not mintable and not burnable.

It is also notable that AERUM Technology used recent contract versions from OpenZeppelin, since these addressed a common race-condition issue regarding *approvals* by implementing the *increase* and *decrease* functions. The extension to the base contracts was implemented in accordance with the best practices and no issues have been found. Overall, the *AerumToken* contract is up to standards and deemed to be secure.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Identified Vulnerabilities

The following sections list both vulnerabilities and implementation issues spotted during the testing period. Note that findings are listed in a chronological order rather than by their degree of severity and impact. The aforementioned severity rank is simply given in brackets following the title heading for each vulnerability. Each vulnerability is additionally given a unique identifier (e.g. *AER-01-001*) for the purpose of facilitating any future follow-up correspondence.

AER-02-001 Token: Pausable not limited to *crowdsale* (*Medium*)

The AERUM *Litepaper* includes a statement about the pausable feature of the XRM token: *"Pausable: yes (transfers are paused until the Token Sale ends)"*. Cure53 found that this is not implemented as described. The *AerumToken* inherits from *PausableToken,* which simply allows the owner to pause and unpause at will, thus making it possible to freeze *XRM* transfers at any time. This means token holders have no guarantees beyond the trust in the AERUM's owner.

Affected Code:

```
contract AerumToken is Ownable, PausableToken {
    string public name = "Aerum";
    string public symbol = "XRM";
[...]
```

As this is a mismatch between marketing material and the actual implementation, it is recommended to either change the marketing material and inform the potential investors or, ideally, implement the described functionality in the smart contract.

Developer Response: *Appropriate measures will be taken by Aerum to maintain XRM token in paused state for duration of the Crowdsale, except short periods when token needs to be credited to specific campaign participants. After the Crowdsale finalized XRM token will be unpaused and Aerum will recede ownership rights in XRM token contract by setting them to 0x0 address*

AER-02-002 ICO: Lack of enforcing for soft and hard caps (*Medium*)

Another claim made by AERUM pertains to the soft cap and hard cap of 5 million and 20 million USD in Ether. While auditing the *crowdsale* contract, it was found that no such caps were implemented. The main challenge is that smart contracts have no trusted source of the price of USD per Ether. While this matter is explored in more depth AER-02-007, the fact of the matter is that even if the contract had a trustworthy source for the USD price, no functionality ensuring the soft and hard caps had been implemented. Actually, the function that seems to implement such a check only looks at the number of

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

tokens and not the USD amount. This means the decision on whether the funding goal has been reached is solely controlled by the owner.

Affected Code:

```
function setGoalReached(bool _success) external onlyOwner {
    goalReached = _success;
}
[...]
function capReached() public view returns (bool) {
    return tokensSold >= token.balanceOf(this);
}
```

It is recommended to add clear information to the public material and inform investors that these caps are not actually contractually enforced. While AERUM could implement a cap based on the USD price, said price is untrusted information controlled by the owner too, so it cannot be resolved in the smart contract's code itself. Alternatively, AERUM could define a fixed Ether to XRM rate when creating the *crowdsale* contract and remove functions that can adjust rates once the *crowdsale* has started.

Developer Response: *Aerum will provide campaign participants with regular updates about the state of Private sale round and will update campaign progress via TokenSale dashboard at the aerum.com landing page combining contributions from public and private sales to determine whether Soft or Hard caps have been reach. The crowdsale contract will be called to update the status of the crowdsale once either cap has been reached. Aerum intends to maintain utmost transparency of the process.*

AER-02-003 ICO: Token vesting schedule not implemented *(Low)*

In the "*Litepaper",* AERUM claims that various token vesting schedules are implemented to gradually unlock tokens. This aims to *"[...] ensure large stakeholders' interests are aligned with the rest of community and they are not able to exert strong pressure on token pricing as tokens are being released with a delay and slowly over a period, [...]"*[1]. However, no such vesting schedule has been found, neither in the *crowdsale*, nor in the token contract.

To not deceive the community, it is recommended to either remove token vesting from the AERUM's promises entirely, or to actually implement this functionality.

Developer Response: *Aerum will place all tokens sold privately into Vesting Wallets, that will stipulate the vesting schedule outlined in Lite and White papers and on the landing page.*

[1] https://aerum.com/en/lite-paper

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

AER-02-004 ICO: Token sale amount not constant *(Low)*

While reviewing how the tokens are allocated for the *crowdsale,* it was found that any XRM (*AerumToken)* holder, such as the owner, can increase the amount of the tokens for the sale at any time. AERUM claims that "*Overall 60% of all issued tokens will be offered at the Token Sale"* and even lists specific quantities. But, as can be seen in the code, the number of the available tokens for the *crowdsale* is determined by the *AerumToken* balance of the *crowdsale* contract. Therefore, if the owner who holds all tokens initially transfers more tokens to the *crowdsale* contract's address, the number of tokens available for sale will increase.

Affected Code

```
function capReached() public view returns (bool) {
    return tokensSold >= token.balanceOf(this);
}
```

It is recommended to initialize the *crowdsale* contract with the specific amount and ensure that it cannot be changed after the fact.

Developer Response: *Aerum will install a procedure where 300 000 000 XRM tokens will be deposited into a Crowdsale contract then all token transfers will be paused. Tokens sent to bounty participants and marketing partners will be able to be transferred to the Crowdsale contract. Aerum will ensures that maximum amount of tokens sold via the Crowdsale contract will not exceeds announced 300,000,000.*

AER-02-005 ICO: Owner can withdraw any Ether or XRM *(High)*

Investigating who has access and control over the raised Ether, as well as the XRM tokens held by the *crowdsale* contract, has demonstrated that the owner has full control over them at all times. Investors have no guarantees about obtaining an Ether refund in case of an unsuccessful *crowdsale.* Despite some checks, the owner can also transfer XRM.

Affected Code:

```
function sendTokens(address _to, uint256 _amount) external onlyOwner {
    if (!hasClosed() || goalReached) {
        // NOTE: if crowdsale not finished or successful
             we should keep at least tokens sold
        _ensureTokensAvailable(_amount);
    }
    token.transfer(_to, _amount);
}
[...]
function setGoalReached(bool _success) external onlyOwner {
```

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

```
    goalReached = _success;
}
[...]
function finalize() public onlyOwner {
    require(!isFinalized);
    // NOTE: We do this because we would like to allow withdrawals earlier
             than closing time in case of crowdsale success
    closingTime = block.timestamp;
    isFinalized = true;
    emit Finalized();
}
```

The code above illustrates that the *sendTokens* function tries to prevent the withdrawal of unclaimed XRM unless the *crowdsale* closing time has passed or the goal has been reached. But the issue is that both of these cases are fully controlled by the owner through the *finalize* and *setGoalReached* functions. So even if the promised caps are reached and the token sale was finalized, the owner could simply set the *goalReached* to *false* and then withdraw all XRM before the buyers rightfully withdraw their tokens.

Affected Code:

```
function ownerWithdraw(uint256 _amount) external onlyOwner {
    require(_amount > 0);
    wallet.transfer(_amount);
    emit OwnerWithdraw(_amount);
}
```

The *ownerWithdraw* function simply allows the owner to send any Ether to the specified wallet at any time. This goes directly against the claims made by AERUM, as is further highlighted in AER-02-006.

It is recommended to implement proper checks to prevent the withdrawals of the already bought tokens and also to stop the possibility of withdrawing funds before the *crowdsale* has been completed. Nevertheless, as AER-02-002 shows, the success or failure of the *crowdsale* is also controlled by the owner, thus this issue cannot be addressed in isolation because the owner could still take out all assets.

Developer Response: *Aerum will install necessary measures and checks to ensure that 300,000,000 XRM will be kept in the Crowdsale contract and will only withdraw unsold tokens after the Crowdsale finalization and sold tokens distribution. Aerum will be transferring out collected Ether from the crowdsale contract on regular basis as a security measure and to be able to hedge Ether price fluctuations. Also Aerum states in the Token Purchase Agreement that a reasonable use of proceeds of both Private and Public sales is accepted for the benefit on Campaign before the soft cap is reached. If*

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

the soft cap is not reached by the end of Crowdsale, the remained funds will be returned back to sale participants in respective proportions to the total amount of funds collected during the Campaign.

AER-02-006 ICO: Collected funds might not be returned *(High)*

In connection to AER-02-005, it should also be highlighted that the AERUM *"Litepaper"* claims that *"If the a soft cap is not reached by the end of all token sale rounds, collected funds will be returned to participants.".* Looking at the actual contract's code proves this statement to be false. Not only can the owner withdraw all funds at any time, investors might only receive a percentage of the initial investment in case AERUM withdraws Ether.

Affected Code:

```
function claimRefund() public {
    require(isFinalized);
    require(!goalReached);

    uint256 refundPercentage = _refundPercentage();
    uint256 amountInvested = weiInvested[msg.sender];
    uint256 amountRefunded =
amountInvested.mul(refundPercentage).div(percentage);
    weiInvested[msg.sender] = 0;
    usdInvested[msg.sender] = 0;
    msg.sender.transfer(amountRefunded);

    emit Refund(msg.sender, amountInvested, amountRefunded);
}
```

The smart contract should reflect the claims made in the information shared with the investors. Once again, either the material needs to be updated or the necessary changes must be made in the code.

Developer Response: *Aerum permits a reasonable use of proceeds of both Private and Public sales is accepted for the benefit on Campaign before the soft cap is reached. If the soft cap is not reached by the end of Crowdsale, the remained funds will be returned back to sale participants in respective proportions to the total amount of funds collected during the Campaign. Aerum will make sure this information is present not only in the Purchase Agreement but also in Lite and White paper to ensure utmost transparency.*

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

AER-02-007 ICO: Ether price in USD Oracle *(Medium)*

As already discussed above, Ethereum smart contracts have no trusted information source regarding the current USD price of Ether. However, the AERUM *crowdsale* contract depends on the price of Ether and makes use of a so-called Oracle as a trusted source of this data. Ideally, this is a third-party that the owner and investors can trust. The problem is that the use of a specific Oracle service cannot be derived from the audit and has not been mentioned in the documentation. Still, the current implementation allows the owner to make most Oracle-related calls anyway and the owner has full control over the address of the Oracle too. This means that the owner has full control over any price information and the role of a trusted Oracle needs to be judged as meaningless.

Affected Code:

```
function setOracle(address _oracle) public onlyOwner {
    oracle = _oracle;
}

[...]

function setRateAndEtherPrice(uint256 _whitelistedRate, uint256 _publicRate,
uint256 _cents) external onlyOwnerOrOracle {
    setRate(_whitelistedRate, _publicRate);
    setEtherPrice(_cents);
}
```

To reiterate, the USD price and, hence, the *crowdsale*'s soft and hard caps cannot realistically be implemented in the smart contract, it is recommended to instead initialize the *crowdsale* with a constant exchange rate and a well-defined cap. A revised approach should automatically trigger a successful funding or deny it otherwise.

Developer Response: *Aerum needs to record amount of funds collected in USD not in Ether to oblige legal requirements of doing KYC/AML for purchases above the certain threshold values. That reason and Ether price volatility do not permit setting a fixed token price at the beginning of the sale. Aerum will implement compensation control measures to ensure that fair pricing information will be fed into the Crowdsale contract that would be beneficial and transparent for buyers.*

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Conclusions

The results of this Cure53 2018 security assessment of the AERUM smart contracts are quite mixed. On the one hand, from a technical standpoint, the security posture of the product is rather excellent. This is because the AERUM compound relies on a number of the already well-audited and solid contracts by OpenZepellin. This makes their efforts to implement the *AerumToken* and *AuerumCrowdsale* securely fruitful. On the other hand, a number of security promises and security-related statements that the AERUM project has made on their website, presumably as means to promote their product to potential users/customers, cannot be proven true and did not held up to the Cure53 auditors' scrutiny.

To give more details, the *AerumToken* follows best practices of implementing an ERC-20 token and no serious issues were found to affect the funds of the token holders. This is clearly a commendable result. However, seven items have been identified and two among them were ranked as "*High"* in terms of the possible negative implications. All seven claims directly relate to the claims made by AERUM in the marketing material. While it might theoretically be argued that the issues do not impact on the security from the point of view of the AERUM as a smart contract issuer, they might have significant bearing on the decisions and outcomes for the actual participants of the *crowdsale*. Several claims highlighted by Cure53 as false are based on the AERUM policies that are documented yet not implemented in the contracts. The participants either need to trust AERUM or cannot participate in the exchanges because no contractual guarantees are present. Thus, from the point of view of an investor, the *AerumCrowdsale* contract acts as just another centralized system that has to be trusted. In other words, it does not give any benefits or assurances about being a smart contract that implements these promises in line with what is stated to the broader public.

It needs to be noted that some of the issues could be resolved in the smart contracts. However, several items are related to practical and regulatory problems that cannot be solved within a smart contract framework. At the same time, AERUM took the live-reported issues very seriously and implemented various steps to mitigate them. This can be seen from the exchanges included in the report, reflecting that the AERUM team has read the initial version of the reports and shared their feedback with Cure53 after an online-meeting. It is hoped that such a comprehensive picture of security, seen as a need to both meet the technical standards, and ascertain that the documentation must be truthful, will help the AERUM team moving forward.

Cure53 would like to thank Patrick O'Sullivan, Alex Randarevich and Petro Sidlovskyy from the AERUM Technology team for their excellent project coordination, support and assistance, both before and during this assignment.