SECURITY AUDIT OF UPPSALA - SENTINEL PROTOCOL SMART CONTRACT

PUBLIC REPORT

MAY 21, 2018

Verichains Lab

info@verichains.io

https://www.verichains.io

Driving Technology >> Forward

EXECUTIVE SUMMARY

This Security Audit Report prepared by Verichains Lab on May 21, 2018. We would like to thank Uppsala Foundation to trust Verichains Lab to audit smart contracts. Delivering high-quality audits is always our top priority.

This audit focused on identifying security flaws in code and the design of the smart contracts. It was conducted on commit *`6778f44d0f9772227b9c8d9913769bf97c26c4d9`* of branch *`master`* from GitHub repository of Uppsala.

Overall, the audited code demonstrates high code quality standards adopted and effective use of modularity and security best practices. No major vulnerabilities were discovered during the audit.

CONFIDENTIALITY NOTICE

This report may contain privileged and confidential information, or information of a proprietary nature, and information on vulnerabilities, potential impacts, attack vectors of vulnerabilities which were discovered in the process of the audit.

The information in this report is intended only for the person to whom it is addressed and/or otherwise authorized personnel of Uppsala. If you are not the intended recipient, you are hereby notified that you have received this document in error, and that any review, dissemination, printing, or copying of this message is strictly prohibited. If you have received this communication in error, please delete it immediately.

CONTENTS

ACRONYMS AND ABBREVIATIONS

Ethereum	An open source platform based on blockchain technology to create and distribute smart contracts and decentralized applications.
ETH (Ether)	A cryptocurrency whose blockchain is generated by the Ethereum platform. Ether is used for payment of transactions and computing services in the Ethereum network.
Smart contract	A computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract.
Solidity	A contract-oriented, high-level language for implementing smart contracts for the Ethereum platform.
Solc	A compiler for Solidity.
EVM	Ethereum Virtual Machine.

AUDIT OVERVIEW

ABOUT UPPSALA TOKENS (UPP)

Token Name	Sentinel Protocol
Symbol	UPP
Decimals	18
Total Tokens	500,000,000 UPP
Available for Token Sale	300,000,000 UPP
Pre-Sentinels	Up to 5,000,000 UPP [concluded]
Private Sale Allocation	168,500,000 UPP (1 ETH = 5,000 UPP plus a 30% bonus) [concluded]
Presale Allocation	87,500,000 UPP (1 ETH = 5,000 UPP plus a 15% bonus) • 43,750,000 UPP for strategic contributors [concluded] • 43,750,000 UPP for public presale [May 23rd – 26th]
Crowdsale	[39,000,000 UPP (1 ETH = 5,000 UPP with no bonus) [May 27th – Jun 10th]

Details for the upcoming Presale and Crowdsale

Presale

- Total : 43,750,000
- Rate : 1:5000 + 750 (15% bonus)
- Individual Cap : Min 30, Max 300
- Gas Limit, Gas Price : 200000, 50 Gwei
- Lock up period : 180 days after presale
- Opening date : 23 May 21:00 (GMT+8)
- Closing date : 26 May 21:00 (GMT+8)

Crowdsale

- Total : 39,000,000
- Rate : 1:5000 (no bonus)
- Individual Cap : Min 0.5, Max 10
- Gas Limit, Gas Price : 200000, 50 Gwei
- Opening date : 27 May 21:00(GMT+8)
- Closing date : 10 June 21:00(GMT+8)

All bonuses have a lock-up period of 180 days after sale. The crowdsale round does not have any bonuses.

SCOPE OF THE AUDIT

This audit focused on identifying security flaws in code and the design of the smart contracts for the upcoming Presale, Crowdsale, and Token. It was conducted on commit *6778f44d0f9772227b9c8d9913769bf97c26c4d9* of branch *master* from GitHub repository of Uppsala.

Repository URL: https://github.com/peppermint2018/uppsala-ico

The scope of the audit is limited to the following 5 source code files received on May 20, 2018:

Source File	SHA256 Hash
Migrations.sol	3512b45aacf3ffa5e901c775696c6869d90519fe8eef06536dede95c19f0c815
UppsalaCrowdsale.sol	13be80a327b453f4703fa4dbec14e858694c47081b0f3d731b9d795aacc1ea03
UppsalaPresale.sol	2de1523d1bcdfd7b4eab793ae10d775b62631208e918314b3c1dd268874e6d83
UppsalaToken.sol	94fae30ff72f4425546f309ef6b8f568f8c33dfee06705ef500f427985e0c4c7
UserMinMaxCrowdsale.sol	3d584886734ecfcf750b5ace850f51149dba9bb03ae4411dfd35f8803579652b

Uppsala's presale, crowdsale and token contracts are based mainly on OpenZeppelin, the industry-standard library with extra features including WhitelistedCrowdsale, CappedCrowdsale, TimedCrowdsale and Pausable.

AUDIT METHODOLOGY

Our security audit process for smart contract includes two steps:

- Smart contract codes are scanned/tested for commonly known and more specific vulnerabilities using public and in-house automated analysis tools.
- Manual audit of the codes for security issues. The contracts are manually analyzed to look for any potential problems.

Following is the list of commonly known vulnerabilities that was considered during the audit of the smart contract:

- Integer Overflow and Underflow
- TimeStamp Dependence
- Race Conditions
- Transaction-Ordering Dependence
- DoS with (Unexpected) revert
- Dos with Block Gas Limit
- Gas Usage, Gas Limit and Loops
- Redundant fallback function
- Unsafe type Inference

- Reentrancy
- Explicit visibility of functions state variables (external, internal, private and public)
- Logic Flaws

For vulnerabilities, we categorize the findings into categories, depending on their criticality:

Low An issue that does not have a significant impact, can be considered as less important

Medium A vulnerability that could affect the desired outcome of executing the contract with medium impact in a specific scenario; needs to be fixed.

High A vulnerability that could affect the desired outcome of executing the contract with high impact; needs to be fixed with high priority.

Critical A vulnerability that can disrupt the contract functioning; creates a critical risk to the contract; required to be fixed immediately.

AUDIT RESULT

VULNERABILITIES FINDINGS

No vulnerabilities found.

RECOMMENDATIONS / SUGGESTIONS

- Use constant instead of hard-coded values for gas price and gas limit, for example:

```
uint256 constant public MAX_GAS_PRICE = 50000000000;
uint256 constant public MAX_GAS = 200000;
```

- Solidity contracts can have a special form of comments that form the basis of the Ethereum Natural Specification Format. Please consider to change the comments inside smart contracts following https://github.com/ethereum/wiki/wiki/Ethereum-Natural-Specification-Format.

CONCLUSION

Uppsala's presale, crowdsale and token smart contracts have been audited by Verichains Lab using analysis tools and manual code review. Overall, the audited code demonstrates high code quality standards adopted and effective use of modularity and security best practices. No major vulnerabilities were discovered during the audit.

LIMITATIONS

All parameters for presale and crowdsale contracts such as number of tokens, rate, bonus, duration... will be passed to the contract constructor when these contracts are created so verifying the correctness of initilized parameters is not possible at the time of this audit.

Please note that security auditing cannot uncover all existing vulnerabilities, and even an audit in which no vulnerabilities are found is not a guarantee for a 100% secure smart contract. However, auditing allows discovering vulnerabilities that were unobserved, overlooked during development and areas where additional security measures are necessary.

APPENDIX I

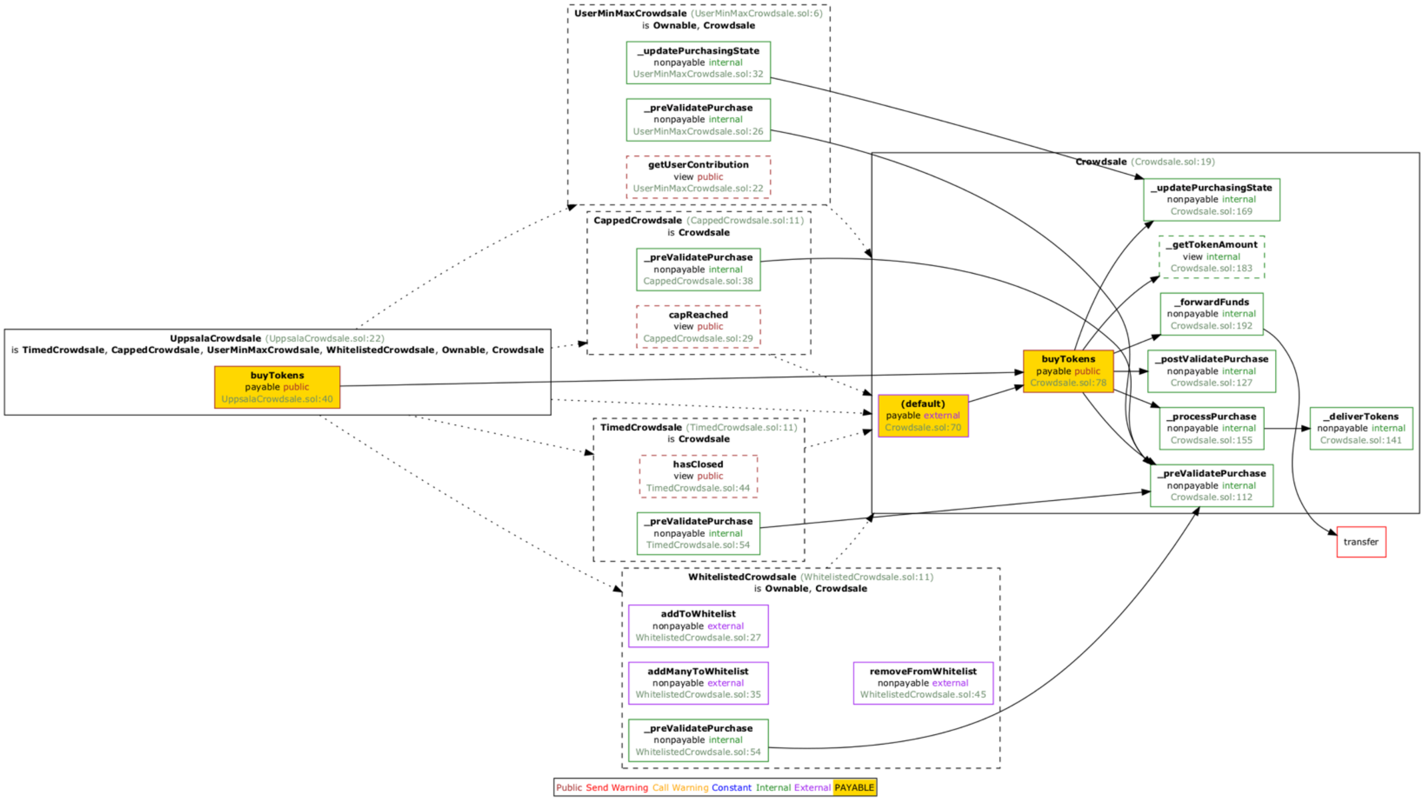

Figure 1 Call graph of UppsalaCrowdsale

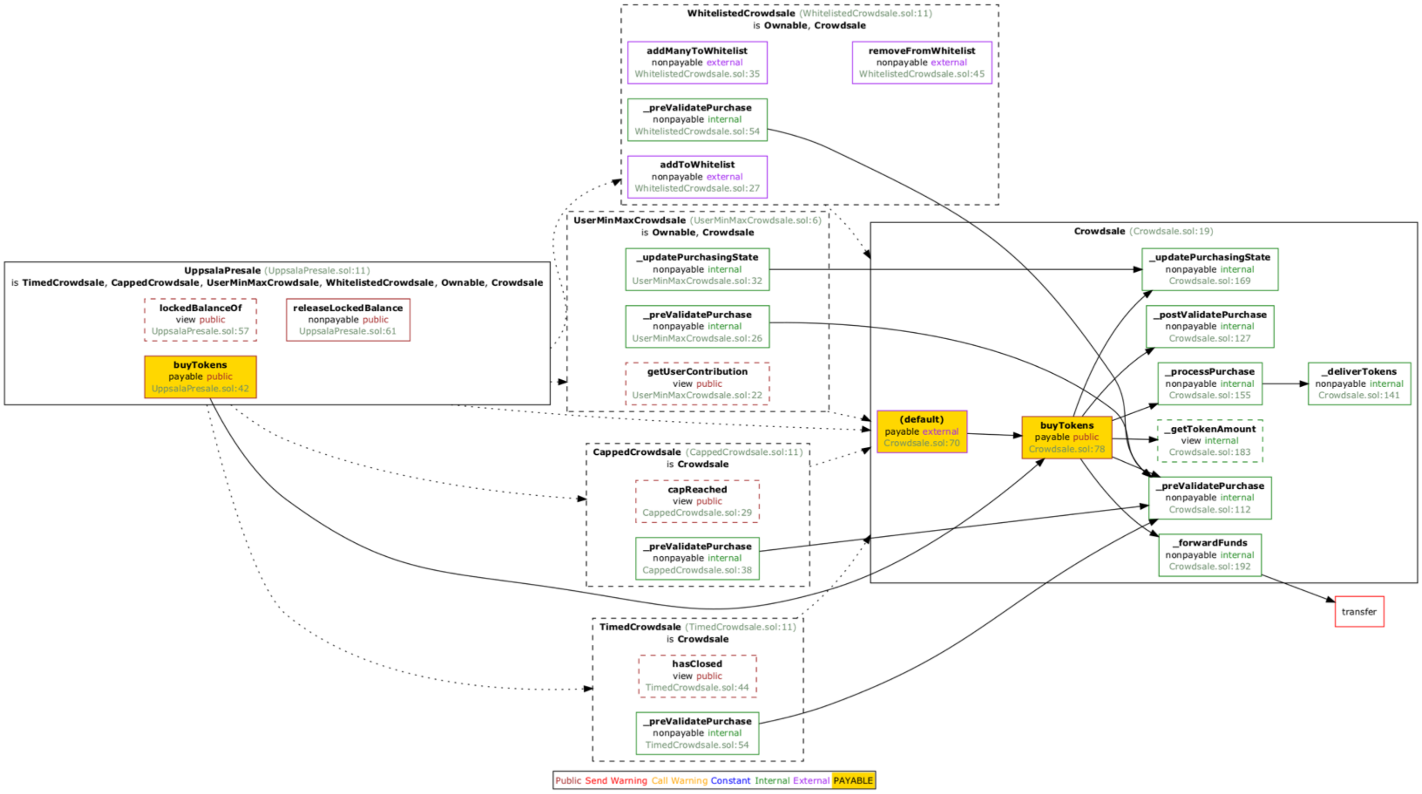

Figure 2 Call graph of UppsalaPresale

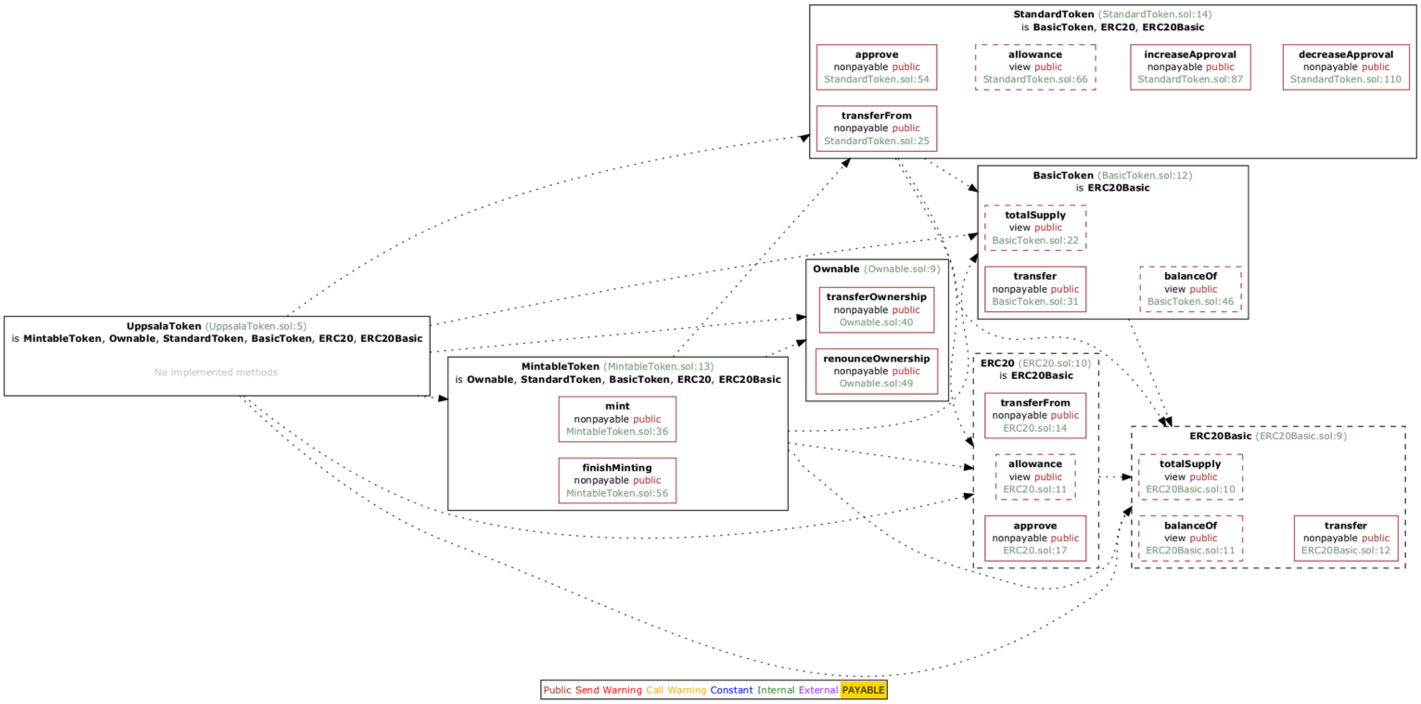

Figure 3 Call graph of UppsalaToken