(https://blog.coinfabrik.com/)

SMART CONTRACT AUDIT (HTTPS://BLOG.COINFABRIK.COM/CATEGORY/SMART-CONTRACTS/SMART-CONTRACT-AUDIT-SMART-CONTRACTS/)

Security Audit – Send (SDT) Token Sale ICO Smart Contract

Ariel Yabo (https://blog.coinfabrik.com/author/ariel-yabo/)

March 6, 2018 (https://blog.coinfabrik.com/smart-contracts/security-audit-send-sdt-token-sale-ico-smart-contract/)

Coinfabrik was asked to audit the smart contracts for the Send Token sale. In the first part, we will give a summary of our discoveries and follow them with the details of our findings.

Send (SDT) is making an ICO to create a 7-day price-stable crypto token that discovers a new market price once a week based on cross-border transaction volume and network liquidity.

Contents

Summary

The audited contracts are from the SendProtocol repository at https://github.com/SendProtocol/sdt-contracts.git. The audit is based on the commit 0e00bbab50d0ebe341b7d54475aceafb6303ff66, and updated to reflect changes at ba9d86e39d644ba3e63b2ed9f5b0417a5c381e20.

The audited contracts are:

- Escrow.sol: Escrow contract for the token
- Polls.sol: Basic polling contract, requires tokens before a certain block to vote.
- SnapshotToken.sol: Implements snapshot capabilities for the token, to allow the poll check balances before a determined block.

- SendToken.sol: Token implementation, inherits both BurnableToken and SnapshotToken.
- SDT.sol: Final SendToken with ERC20 parameters set.
- SaleProxy.sol: Acts as a proxy for the Token Sale.
- TokenSale.sol: ICO implementation.
- TokenVesting.sol: Allows the ICO to vest tokens.
- IEscrow.sol: Escrow interface
- ISendToken.sol: SendToken interface
- ISnapshotToken.sol: SnapshotToken interface
- ITokenSale.sol: TokenSale interface

These checks are performed:

- Misuse of the different call methods: call.value(), send() and transfer().
- Integer rounding errors, overflow, underflow and related usage of SafeMath functions.
- Old compiler version pragmas.
- Race conditions such as reentrancy attacks or front-running.
- Misuse of block timestamps, assuming things other than them being strictly increasing.
- Contract soft locking attacks (DoS).
- Potential gas cost of functions being over the gas limit.
- Missing function qualifiers or misuse of them.
- Fallback functions with higher gas cost than a what a transfer or send call allows.
- Underhanded or erroneous code.
- Code and contract interaction complexity.
- Wrong or missing error handling.
- Overuse of transfers in a single transaction instead of using withdrawal patterns.
- Insufficient coverage of function input requirements.
- Standard non-compliance (EIP20, etc)

Detailed findings

Critical severity

Unverified BTC purchases allowed through SaleProxy

The *TokenSale* contract only allows contributions that go through a *SaleProxy* contract. However, the *SaleProxy* contract itself doesn't provide any sort of check that a pertinent BTC transfer has been included in the bitcoin blockchain. We urge the developers to review the design of this feature if contributions through BTC are intended.

A simple example attack would be as follows:

1. Address A gets whitelisted for contributions in the token sale contract.
2. A signs a transaction with data corresponding to *btcPurchase(address _beneficiary, uint256 _btcValue)* ABI encoding found in the *SaleProxy* contract. *_btcValue* can be any amount. In particular, it needs to be bigger than $10 once converted so it doesn't get rejected by the contract.
3. The token sale contract assigns vested tokens to A depending on the vesting configuration of the *SaleProxy* contract used by A.

Fixed in commit 392d93c0a7109a86fba7de96d3137c787b70f842

Minor severity

(https://blog.coinfabrik.com/)

Not extensive SafeMath usage

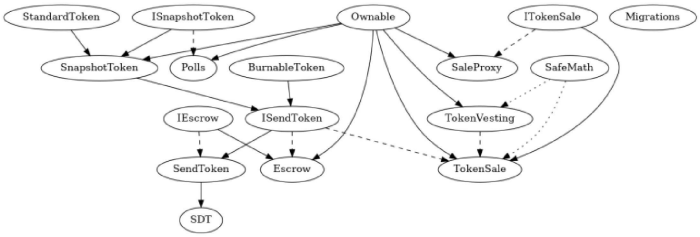
(https://blog.coinfabrik.com/)

Some contracts like *TokenSale.sol*, are missing *SafeMath* functions for sensible calculations. Consider extending the usage of *SafeMath* to these contracts.

Fixed in commit aba8cf827774ce0b7abc8254271b8efe3c9761c4

Enhancements

Unnecessary and misleading interfaces



There are four interface files: *ISendToken*, *ISnapshotToken*, *ITokenSale*, *IEscrow*. These are declared as contracts instead of being just interfaces. In addition, *ISendToken* inherits from two contracts. If the intention is to only use these interfaces inside the project, we recommend removing these contracts, and instead directly use the corresponding contract which the interface is derived from, as it is less error prone and much simpler. Consider using internal for those functions you don't want to expose to other contracts. If not, we recommend using the interface directive to declare an interface, and moving the dependencies to the corresponding contract.

Fixed in commit d313e021aaaa0c9b9021207bd3e168a1545971b2

Conclusion

Overall the contracts were well documented and had extensive background documentation of their development. The main idea behind the contracts is simple and the deployment flow is expectable. It was also a good idea to reuse well tested OpenZeppelin contracts.

Do you want to know what is Coinfabrik Auditing Process?

Check our A-Z Smart Contract Audit Guide (https://blog.coinfabrik.com/smart-contract-audits-ultimate-security-guide/) or you could request a quote (https://www.coinfabrik.com#contact_us) for your project.

Related Posts

(https://blog.coinfabrik.com/smart-

contracts/smart-contract-audit-smart-contracts/dmtoken-token-sale-smart-contract-audit/)

(https://blog.coinfabrik.com/)

DMToken Token Sale Smart Contract Audit (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/dmtoken-token-sale-smart-contract-audit/)

Coinfabrik was hired to audit the contract in terms of its security. First of all,...

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rightmesh-token-sale-smart-contract-audit-master/)

Rightmesh Token Sale Smart Contract Audit (Master) (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rightmesh-token-sale-smart-contract-audit-master/)

Coinfabrik was asked to audit the contracts for the RightMesh Token sale. In the first...

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/bricks4us-token-smart-contract-

Bricks4US Token Smart Contract Security Audit (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/bricks4us-token-smart-contract-security-audit/)

CoinFabrik was asked to audit the contracts for the Bricks4Us token sale. Firstly, we will...

security-

audit/)

(https://blog.coinfabrik.com/)

(https://blog.coinfabrik.com/)

Tags:

SDT
(https://blog.coinfabrik.com/tag/sdt/)

Send
(https://blog.coinfabrik.com/tag/send/)

smart-contracts
(https://blog.coinfabrik.com/tag/smart-contracts/)

Token
(https://blog.coinfabrik.com/tag/token/)

SHARE ON

(https://www.facebook.com/sharer/sharer.php?u=https://blog.coinfabrik.com/smart-contracts/security-audit
send-sdt-token-sale-ico-smart-contract/)
(https://twitter.com/intent/tweet?text=Security%20Audit%20–
%20Send%20(SDT)%20Token%20Sale%20ICO%20Smart%20Contract&url=https://blog.coinfabrik.com/smart-
contracts/security-audit-send-sdt-token-sale-ico-smart-contract/)
(https://pinterest.com/pin/create/button/?url=&media=https://blog.coinfabrik.com/wp-
content/uploads/2018/03/send-
STN.png&description=Security+Audit+%26%238211%3B+Send+%28SDT%29+Token+Sale+ICO+Smart+Contra
(https://www.linkedin.com/shareArticle?mini=true&url=https://blog.coinfabrik.com/smart-contracts/security-a
send-sdt-token-sale-ico-smart-contract/&title=Security%20Audit%20–
%20Send%20(SDT)%20Token%20Sale%20ICO%20Smart%20Contract&source=CoinFabrik%20Blog)

← PREVIOUS ARTICLE

Rightmesh Token Sale Smart Contract Audit (Master) (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rightmesh-token-sale-smart-contract-audit-master/)

NEXT ARTICLE →

RCN Smart Contracts Audit v2 (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/rcn-smart-contracts-audit-v2/)

You may also like

(https://blog.coinfabrik.com/smart-contracts/magic-bridge-audit/)

Magic Bridge Audit (https://blog.coinfabrik.com/smart-contracts/magic-bridge-audit/)

(https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/mintingfactoryv2-baseupgradablemarketplace-kodav3upgradablegatedmarketplace/)

2 months ago

Smart Contract Audit (https://blog.coinfabrik.com/category/smart-contracts/smart-contract-audit-smart-contracts/)

MintingFactoryV2, BaseUpgradableMarketplace & KODAV3UpgradableGatedMarketplace (https://blog.coinfabrik.com/smart-contracts/smart-contract-audit-smart-contracts/mintingfactoryv2-

baseupgradablemarketplace-kodav3upgradablegatedmarketplace/)
(https://blog.coinfabrik.com/)

(https://blog.coinfabrik.com/)

(https://blog.coinfabrik.com/feed/)

(https://ar.linkedin.com/company/coinfabrik)

(https://twitter.com/coinfabrik)

(https://www.youtube.com/channel/UC2GmjCr7aEz-il31kqOy9aw)

(https://www.facebook.com/CoinFabrik/)

(https://www.reddit.com/r/CoinFabrik/)

(https://github.com/coinfabrik)

Made with ♥ in Buenos Aires, Argentina.