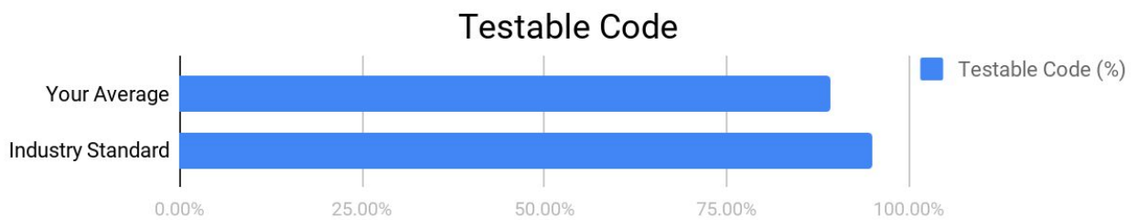
DCORP BV Contract Audit

by Hosho, April 2018

Executive Summary

This document outlines the overall security of DCORP BV's smart contract as evaluated by Hosho's Smart Contract auditing team. The scope of this audit was to analyze and document DCORP BV's contract codebase for quality, security, and correctness.

All issues have been remediated. (See Complete Analysis)



Testable code is 89.21% which is lower than industry standard but not due to any issues. (See Coverage Report)

It should be noted that this audit is not an endorsement of the reliability or effectiveness of the contract, rather limited to an assessment of the logic and implementation. In order to ensure a secure contract that's able to withstand the Ethereum network's fast-paced and rapidly changing environment, we at Hosho recommend that the DCORP BV Team put in place a bug bounty program to encourage further and active analysis of the smart contract.

Table Of Contents

1. Auditing Strategy and Techniques Applied

The Hosho Team has performed a thorough review of the smart contract code, the latest version as written and updated on April 3, 2018. All main contract files were reviewed using the following tools and processes. (See All Files Covered)

Throughout the review process, care was taken to ensure that the contract:

- Documentation and code comments match logic and behavior;
- Follows best practices in efficient use of gas, without unnecessary waste; and
- Uses methods safe from reentrance attacks.
- Is not affected by the latest vulnerabilities
- The included token implements and adheres to existing ERC-20 Token standards appropriately and effectively;

The Hosho Team has followed best practices and industry-standard techniques to verify the implementation of DCORP BV's contract. To do so, reviewed line-by-line by our team of expert pentesters and smart contract developers, documenting any issues as they were discovered. Part of this work included writing a unit test suite using the Truffle testing framework. In summary, our strategies consist largely of manual collaboration between multiple team members at each stage of the review:

1. Due diligence in assessing the overall code quality of the codebase.
2. Cross-comparison with other, similar smart contracts by industry leaders.
3. Testing contract logic against common and uncommon attack vectors.
4. Thorough, manual review of the codebase, line-by-line.
5. Deploying the smart contract to testnet and production networks using multiple client implementations to run live tests.

2. Structure Analysis and Test Results

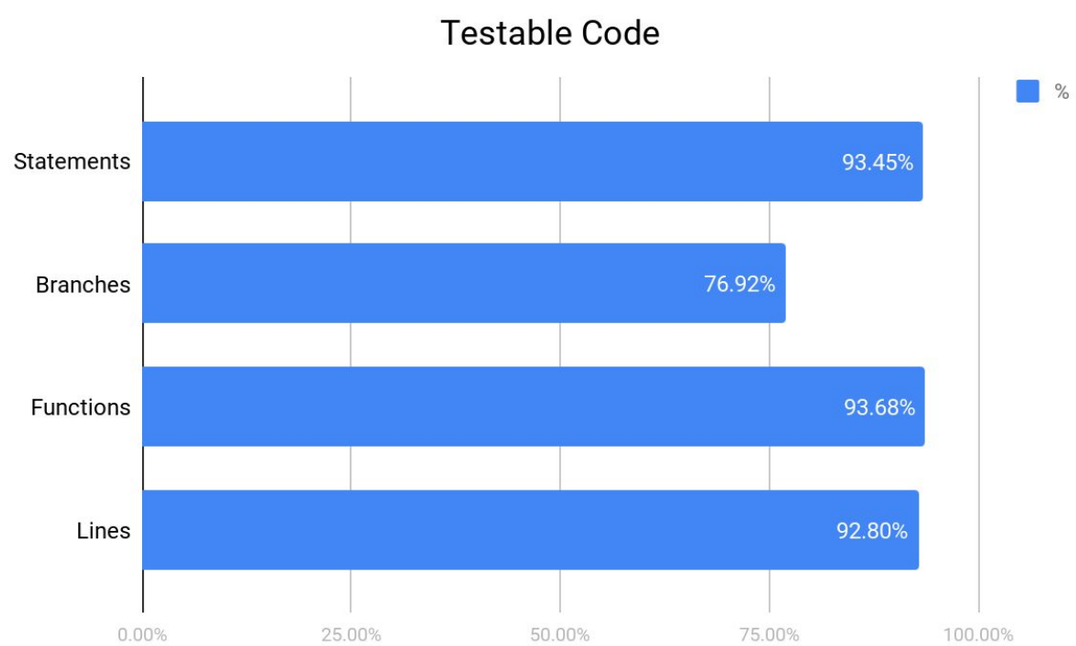
2.1. Summary

The DCORP BV contracts have been created to operate as an interim wallet with a generated simple password management system. As it is part of a larger ecosystem, these contracts will only interact with small amounts of tokens, that should not require the setup of a platform such as Metamask. Tokens of significant amounts will be protected by 2FA, as determined by the platform.

There were original concerns about these contracts that have been resolved by acknowledging the ecosystem that they exist in through discussions with the DCORP BV Team.

2.2 Coverage Report

As part of our work assisting DCORP BV in verifying the correctness of their contract code, our team was responsible for writing a unit test suite using the Truffle testing framework.



For individual files see Additional Coverage Report

2.3 Failing Tests

No failing tests.

See Test Suite Results for all tests.

3. Complete Analysis

For ease of navigation, sections are arranged from most critical to least critical. Issues are tagged "Resolved" or "Unresolved" depending on whether they have been fixed or addressed. Furthermore, the severity of each issue is written as assessed by the risk of exploitation or other unexpected or otherwise unsafe behavior:

- **Informational** - The issue has no impact on the contract's ability to operate.
- **Low** - The issue has minimal impact on the contract's ability to operate.
- **Medium** - The issue affects the ability of the contract to operate in a way that doesn't significantly hinder its behavior.
- **High** - The issue affects the ability of the contract to compile or operate in a significant way.
- **Critical** - The issue affects the contract in such a way that funds may be lost, allocated incorrectly, or otherwise result in a significant loss.

1.1. Resolved, High: Remote Contract Execution

MemberAccount

Explanation

Due to the usage of _target.call.value(), if the short address protection is incorrectly implemented on the remote contract, this code will not be able to execute the _data call against it properly. This is caused by known issues within the implementation of .call() and should be reviewed if needed.

Resolution

Discussions with the DCORP BV Team have revealed that this system will only be interacting with contracts that have been specifically designed to interact with the contracts in this codebase, eliminating this issue.

1.2. Resolved, High: Password Reuse

MemberAccount

Explanation

Each time the `authenticate` modifier is called, it takes the current passphrase and the hash of a new passphrase. It then checks the hash of the the current passphrase against the stored passphrase, before storing the new passphrase into the system. If a passphrase is reused, both the plain text and hash are available, and stored, as part of the transaction. From there they can be used to make the next call.

Additionally, if the transaction fails, the system will revert and not store the newly provided hash. This is problematic, in that the plain text passphrase will have been stored as part of the transaction, which is accessible by anyone.

Resolution

The passwords will be hashed by an external systems. As these hashes will be derived with information from the network itself, the passwords will be unique. The DCORP BV Team verified that the implementation will be taken care of in the dApp, and as such, this is no longer an issue.

1.3. Resolved, Medium: No Password Recovery

Explanation

There is no functionality within the contract that allows for password recovery as is typical, and expected, with password management systems.

Resolution

This functionality is as intended, as it is not designed to interact with other systems except for those designed by the DCORP BV Team. All users will also be notified of this to clarify the functionality, which mitigates this issue.

1.4. Resolved, Medium: Potential Limitation

DCorpAccounts

Explanation

The `onTokensDeposited` function requires the `repository` and `to` parameters to be the same. If the desired functionality is keeping track of account addresses in the repository, which it appears to be as it is not an intended recipient or store of tokens, then this restriction creates an unintended limitation if either piece of information is not present.

Resolution

This part of the system has been created in expectation of the next project, which contains a voting system. This system will interact with `onTokensDeposited` as intended, making this issue resolved by future development according to the DCORP BV Team.

1.5. Resolved, Medium: Possible Incorrect Observer Results

DCorpMemberAccount

Explanation

The `getObserver` function always returns 0x0 for accounts created through the account factory in DCorpAccounts. The only path that returns a different value is in the instance of `memberAccountCode`. This may be as intended but there is no documentation to verify against.

Resolution

There will be a separate contract that sets the observer to be utilized within these contracts. Until that contract is interacting with these, it is expected that the value will be 0x0. This is therefore operating as intended and resolves this issue.

1.6. Resolved, Low: Inconsistent Notification

DCorpAccounts

Explanation

The repository accounts, from DCorpMemberAccounts.sol, notify the function `onTokensWithdrawn` but not `onTokensDeposited`. As the `onTokensDeposited` function is internal, it should be notifying itself.

Result:

The DCORP BV Team has indicated that this is functioning as intended.

---

4. Closing Statement

We are grateful to have been given the opportunity to work with the DCORP BV Team.

The team of experts at Hosho, having backgrounds in all aspects of blockchain, cryptography, and cybersecurity, can say with confidence that the DCORP BV contract is free of any critical issues.

The statements made in this document should not be interpreted as investment or legal advice, nor should its authors be held accountable for decisions made based on them.

We at Hosho recommend that the DCORP BV Team put in place a bug bounty program to encourage further analysis of the smart contract by other third parties.

5. Test Suite Results

Coverage Report:

Contract: ERC-20 Tests for ManagedToken

✓ Should deploy a token with the proper configuration (76ms)

✓ Should allocate tokens per the minting function, and validate balances (167ms)

✓ Should transfer tokens from 0xd86543882b609b1791d39e77f0efc748dfff7dff to 0x42adbad92ed3e86db13e4f6380223f36df9980ef (55ms)

✓ Should not transfer negative token amounts

✓ Should not transfer more tokens than you have

✓ Should allow 0xa3883a50d7d537cec8f9bad8e8404aa8ff3078f3 to authorize 0x341106cb00828c87cd3ac0de55eda7255e04933f to transfer 1000 tokens (43ms)

✓ Should allow 0xa3883a50d7d537cec8f9bad8e8404aa8ff3078f3 to zero out the 0x341106cb00828c87cd3ac0de55eda7255e04933f authorization

✓ Should allow 0x667632a620d245b062c0c83c9749c9bfadf84e3b to authorize 0x53353ef6da4bbb18d242b53a17f7a976265878d5 for 1000 token spend, and 0x53353ef6da4bbb18d242b53a17f7a976265878d5 should be able to send these tokens to 0x341106cb00828c87cd3ac0de55eda7255e04933f (120ms)

✓ Should not allow 0x53353ef6da4bbb18d242b53a17f7a976265878d5 to transfer negative tokens from 0x667632a620d245b062c0c83c9749c9bfadf84e3b

✓ Should not allow 0x53353ef6da4bbb18d242b53a17f7a976265878d5 to transfer tokens from 0x667632a620d245b062c0c83c9749c9bfadf84e3b to 0x0

✓ Should not allow 0x53353ef6da4bbb18d242b53a17f7a976265878d5 to transfer more tokens than authorized from 0x667632a620d245b062c0c83c9749c9bfadf84e3b

✓ Should allow an approval to be set, then increased, and decreased (38ms)

✓ Should permit token burning (84ms)

✓ Should not permit burning of more tokens than owned

ManagedToken tests

✓ Should allow owner to lock token (55ms)

✓ Should require token to be unlocked for transfer (128ms)

Contract: DCorpAccounts

✓ Should start with correct number of accounts

✓ Should allow creating account (50ms)

✓ Should require valid token to update account weight (122ms)

✓ Should require account weight to match token balance (254ms)

✓ Should require account to have been created through the contract

DCorpMemberAccount

✓ Should allow node to add lock to account (67ms)

✓ Should require account to be unlocked before adding lock (104ms)

✓ Should require account owner to remove lock on account

✓ Should allow authenticated caller to withdraw eth (157ms)

✓ Should require successful eth transfer to log event (145ms)

✓ Should require min ether withdraw amount (141ms)

✓ Should allow eth withdraw by authorized account without fee (169ms)

✓ Should allow authenticated caller to withdraw tokens (390ms)

✓ Should allow token withdraw by authorized account without fee (473ms)

✓ Should require min token withdraw amount (221ms)

✓ Should require successful token withdraw to log event (259ms)

✓ Should require token withdrawal to notify contract (297ms)

✓ Should require execute target to be another contract instance (126ms)

✓ Should require execute target be validated through (167ms)

✓ Should require execute to revert on bad return from target call (344ms)

✓ Should allow authenticated caller to execute through target (309ms)

Authentication

✓ Should require correct raw password for authentication (190ms)

✓ Should require new password hash to not be empty (139ms)

✓ Should require correct sender account for 2fa (176ms)

✓ Should require lock to authenticate (54ms)

✓ Should require sender to be lock holder to authenticate (118ms)

✓ Should allow resetting password (172ms)

✓ Should allow authenticated caller to enable and disable 2fa (146ms)

DCorpMemberAccountShared

✓ Should allow owner to set min token withdraw

✓ Should allow owner to set withdraw fee (60ms)

✓ Should allow node to modify withdraw fee (148ms)

✓ Should allow owner to update node (110ms)

✓ Should allow owner to modify lock variables (39ms)

Contract: Observable

✓ Should require owner to register observer (137ms)

✓ Should require owner to unregister observer (52ms)

✓ Should require observer to be owner of tokens to register (45ms)

✓ Should require contract to be of type observer to register or unregister (52ms)

✓ Should have correct observer state (163ms)

Contract: Ownership

MemberAccount Ownership

✓ Should report ownership as sender (50ms)

✓ Should require owner to add node (62ms)

Token Ownership

✓ Should require starting with single owner

✓ Should allow owner to add owners (84ms)

✓ Should allow owner to remove owners (105ms)

6. All Contract Files Tested

Commit Hash: b3176f388962dcdd0ecee083985f9ea897cbdcd5

File	Fingerprint (SHA256)
contracts/infrastructure/behaviour/observer/IObservable.sol	c02a8978c53e2377bc29fb8edcad85a4c67cf3dd84345e9bfcd150a20dd61307
contracts/infrastructure/behaviour/observer/IObserver.sol	71cc4947bfc2ea83fb62495b53492c3f1cd384746f12b6c4a28d11a24d81bc7f
contracts/infrastructure/behaviour/observer/Observable.sol	54fdd99d996bc1ecfeaf63b0decc91853a56aac53a92fbcc5fa8bdd456f2ab57
contracts/infrastructure/behaviour/observer/Observer.sol	502c7187b0b80c6a3a4f21a4711f937b9974f95abcbca7534704e5fb785c4e41
contracts/infrastructure/behaviour/state/IDeployable.sol	38a8c99cc2e3c762179aeb92bcebd31020828eb7b2e547a29821b6027b9fc5e3
contracts/infrastructure/dispatcher/Dispatchable.sol	a44a92ed6f6d569bbacc57e190f50634d087dfe603bcd20d08348df8ae6f1151
contracts/infrastructure/dispatcher/SimpleDispatcher.sol	6a2931eab63f74db72bfb3077ff582641a8779aaea8e6f5b16b03ab6be51be61
contracts/infrastructure/modifier/InputValidator.sol	e9bca32a2fa6f62a1df0513ab5798320dfee9c6f3dae3ecc59556e24d41a6a0f
contracts/infrastructure/ownership/IMultiOwned.sol	76f78b9466dda9b9c2aa4c742f646e88b7136ce2a1b42ea284edaa3e618f246b
contracts/infrastructure/ownership/IOwnership.sol	3416ac9b4798fa171e36b501485e3788c5c46a3cffe13d9bca41e1c4ac7353d9
contracts/infrastructure/ownership/ITransferableOwnership.sol	00fbcbd9335dfdca40561d1ab8aa2978364cd0217a7f433c837a6536764ec20f
contracts/infrastructure/ownership/MultiOwned.sol	8a55695222d9a99e2b550bf9444dd047331486490ef8ac19e6e4bb297ef1ef8b
contracts/infrastructure/ownership/Ownership.sol	3e33ba99fc091df364146aab63bbb3525e07c02c4b98f62aaab9de2187001814
contracts/infrastructure/ownership/TransferableOwnership.sol	c5c764b052669e88867424da255f47657fc0dd30eb67f3041e8f8c4ba33461ef
contracts/source/DCorpAccounts.sol	78f9763ee4bc8b7fb65b314f7872789d0cb7e287897fcff98a160bdd92efc80a

contracts/source/DCorpMemberAccount.sol	19bac5ef5e4b5ffeaac8ecdef3206e5c4a91590febcc797413d10906cb76ce41
contracts/source/DCorpMemberAccountDispatcher.sol	351333d4a50b48f0b1380b27ad65a89251bad27c07bafd281980e384f0cdc573
contracts/source/DCorpMemberAccountShared.sol	d6e5b12484b673ffb5280715433589f4383c967b83c5061eaa61c8002836fd87
contracts/source/IDCorpMemberAccount.sol	dedf2eae90018797846542e1ea4d878645bab4a355eb9dd23a506ce7d42f81d5
contracts/source/account/IAccount.sol	39459c03d79bf123c9670a99bf228c3ce08effd2ae385e1c8e3e05be78e1f079
contracts/source/account/IEtherAccount.sol	ea25824c1a6d9db7a7e3ec7ecc28c3a8d222c33c3ab0714bd4119a6f27ae36db
contracts/source/account/ITokenAccount.sol	0f4e46885127aef90acc5c67b4e97679ac4ce9d023cbadbabcecfcbb8a26763f
contracts/source/account/member/IMemberAccount.sol	5980e70ab18b3375e6d50bed768f582cc53f9f8535c59d4fbd98818e84cb72ec
contracts/source/account/member/IMemberAccountShared.sol	e8be0813d7530a1898a3a76bf6fdb63f5ed3f2c392ce19bf0fe69378dfab44fc
contracts/source/account/member/MemberAccount.sol	222d575b2e2a663d76cf635dea4702d454356dcdbc137685356dbfe0c0fb93ab
contracts/source/account/member/MemberAccountShared.sol	5661bc9c86d3a973d374e20fa04122ec820c86544c84f176f76a263a7df63e96
contracts/source/token/IManagedToken.sol	4cd20aa9a6c930e53d881612902ee5fee129ad7f3664cfb6081a3c0cb268d380
contracts/source/token/IToken.sol	683f9b4a610f11e27dac9013d7157922450d763c01ba3a04e4bcfb06b8eb55e2
contracts/source/token/ManagedToken.sol	b5056108158dea06fff52ee0a732a616c4900009d440a75cbc652494e8e799fb
contracts/source/token/Token.sol	0c80ad96530dfe654d16e5790e9aa95ce639b48c71a87f354c6160ee71b436b9
contracts/source/token/observer/ITokenObserver.sol	12845227ae03bc5ee5a7f022844ef1bb72b1e9400c16dc77aecddfd064a22e20
contracts/source/token/observer/ITokenRepositoryObserver.sol	03096a16284ecedc9aee225a03bc7ba14829b679c54c1825bc394bc4e82d2765
contracts/source/token/observer/TokenObserver.sol	449458d8f65fd3f4a7d03d5a09c22e01a0c174a0c43cc4e656112f3574b11460
contracts/source/token/observer/TokenRepositoryObserver.sol	7cc652a7d805e5c0a06123a4cd6b8bacca42ad81d8957830ce6fde0f9e816694

7. Individual File Coverage Report

File	% Statements	% Branches	% Functions	% Lines
infrastructure/behaviour/observer/IObservable.sol	100.00%	100.00%	100.00%	100.00%
infrastructure/behaviour/observer/IObserver.sol	100.00%	100.00%	100.00%	100.00%
infrastructure/behaviour/observer/Observable.sol	100.00%	87.50%	100.00%	100.00%
infrastructure/behaviour/observer/Observer.sol	0.00%	100.00%	0.00%	0.00%
infrastructure/behaviour/state/IDeployable.sol	100.00%	100.00%	100.00%	100.00%
infrastructure/dispatcher/Dispatchable.sol	100.00%	100.00%	100.00%	100.00%
infrastructure/dispatcher/SimpleDispatcher.sol	100.00%	100.00%	100.00%	100.00%
infrastructure/modifier/InputValidator.sol	100.00%	50.00%	100.00%	100.00%
infrastructure/ownership/IMultiOwned.sol	100.00%	100.00%	100.00%	100.00%
infrastructure/ownership/IOwnership.sol	100.00%	100.00%	100.00%	100.00%
infrastructure/ownership/ITransferableOwnership.sol	100.00%	100.00%	100.00%	100.00%
infrastructure/ownership/MultiOwned.sol	100.00%	100.00%	100.00%	100.00%
infrastructure/ownership/Ownership.sol	100.00%	100.00%	100.00%	100.00%
infrastructure/ownership/TransferableOwnership.sol	100.00%	100.00%	100.00%	100.00%
source/DCorpAccounts.sol	85.37%	53.85%	92.86%	86.96%
source/DCorpMemberAccount.sol	83.33%	100.00%	80.00%	83.33%

source/DCorpMemberAccountDispatcher.sol	100.00%	100.00%	100.00%	100.00%
source/DCorpMemberAccountShared.sol	100.00%	100.00%	100.00%	100.00%
source/IDCorpMemberAccount.sol	100.00%	100.00%	100.00%	100.00%
source/account/IAccount.sol	100.00%	100.00%	100.00%	100.00%
source/account/IEtherAccount.sol	100.00%	100.00%	100.00%	100.00%
source/account/ITokenAccount.sol	100.00%	100.00%	100.00%	100.00%
source/account/member/IMemberAccount.sol	100.00%	100.00%	100.00%	100.00%
source/account/member/IMemberAccountShared.sol	100.00%	100.00%	100.00%	100.00%
source/account/member/MemberAccount.sol	89.74%	81.58%	91.67%	86.36%
source/account/member/MemberAccountShared.sol	98.15%	88.46%	100.00%	96.49%
source/token/IManagedToken.sol	100.00%	100.00%	100.00%	100.00%
source/token/IToken.sol	100.00%	100.00%	100.00%	100.00%
source/token/ManagedToken.sol	95.45%	75.00%	90.00%	95.65%
source/token/Token.sol	100.00%	80.00%	100.00%	100.00%
source/token/observer/ITokenObserver.sol	100.00%	100.00%	100.00%	100.00%
source/token/observer/ITokenRepositoryObserver.sol	100.00%	100.00%	100.00%	100.00%
source/token/observer/TokenObserver.sol	0.00%	100.00%	0.00%	0.00%
source/token/observer/	100.00%	50.00%	100.00%	100.00%
All files	**93.45%**	**76.92%**	**93.68%**	**92.80%**