

Paxos

Date	November 2020
Auditors	Steve Marx

1 Executive Summary

In November 2020, we conducted a security assessment of Paxos's multisig wallet contract. This wallet is based on [Christian Lundkvist's SimpleMultiSig contract](#), which we [previously reviewed](#).

Paxos's modification allows the set of owners to be changed after the wallet is deployed. This report focuses on the impact of those changes.

We performed this assessment between November 7th and November 10th, 2020. The engagement was primarily conducted by Steve Marx. The total effort expended was 8 person-hours.

1.1 Scope

File Name	SHA-1 Hash
SimpleMultiSig.sol	80d54d79fa1ec6268ad42d01f393417edb47bdc5

2 Recommendations

2.1 Update to a more recent version of the Solidity compiler



Resolution

The Solidity version was upgraded to 0.6.11 in [paxosglobal/simple-multisig#8](#). This is the latest version supported by Slither.

As a general best practice, we recommend updating to the latest version of the Solidity compiler. A recent compiler version would also enable a few small improvements, which are listed as separate recommendations.

2.2 Convert DOMAIN_SEPARATOR to be immutable ✓ Fixed

Resolution

This has been fixed in [paxosglobal/simple-multisig#9](#).

Starting with version 0.6.5 of the Solidity compiler, state variables can be marked as `immutable`. Such state variables must be initialized in the contract's constructor. Otherwise they function much like constants. This is a good fit for the `DOMAIN_SEPARATOR`, which is computed at runtime to include the contract's address but otherwise acts as a constant.

code/contracts/SimpleMultiSig.sol:L25

```
bytes32 DOMAIN_SEPARATOR; // hash for EIP712, computed from contract address
```

2.3 Convert the assembly call to Solidity ✓ Fixed

Resolution

This has been fixed in [paxosglobal/simple-multisig#9](#).



Starting with version 0.5.0, the Solidity `address.call()` function no longer has the padding bug described in <https://github.com/ethereum/solidity/issues/2884>. This means it's possible to get rid of the assembly block in `execute()` and instead use Solidity. This is a small win for readability.

code/contracts/SimpleMultiSig.sol:L79-L84

```
// If we make it here all signatures are accounted for.  
// The address.call() syntax is no longer recommended, see:  
// https://github.com/ethereum/solidity/issues/2884  
nonce = nonce + 1;  
bool success = false;  
assembly { success := call(gasLimit, destination, value, add(data, 0x20), m)
```

2.4 Update comments about state mutability ✓ Fixed

Resolution

This has been fixed in [paxosglobal/simple-multisig@3824608](#).

Comments accompanying the `isOwner` and `ownersArr` state variables indicate that they're immutable, but in the modified version of the contract, both can be changed after deployment.

code/contracts/SimpleMultiSig.sol:L22-L23

```
mapping (address => bool) isOwner; // immutable state  
address[] public ownersArr;        // immutable state
```

3 Findings

Each issue has an assigned severity:



Minor issues are subjective in nature. They are typically suggestions around best practices or readability. Code maintainers should use their

own judgment as to whether to address such issues.

- **Medium** issues are objective in nature but are not security vulnerabilities. These should be addressed unless there is a clear reason not to.
- **Major** issues are security vulnerabilities that may not be directly exploitable or may require certain conditions in order to be exploited. All major issues should be addressed.
- **Critical** issues are directly exploitable security vulnerabilities that need to be fixed.

3.1 Owners can never be removed **Critical** **✓ Fixed**

Resolution

This has been fixed in [paxosglobal/simple-multisig#5](#), and appropriate tests have been added.

Description

The intention of `setOwners()` is to replace the current set of owners with a new set of owners. However, the `isOwner` mapping is never updated, which means any address that was ever considered an owner is permanently considered an owner for purposes of signing transactions.

Recommendation

In `setOwners_()`, before adding new owners, loop through the current set of owners and clear their `isOwner` booleans, as in the following code:

```
for (uint256 i = 0; i < ownersArr.length; i++) {  
    isOwner[ownersArr[i]] = false;  
}
```

Appendix 1 - Disclosure



ConsenSys Diligence ("CD") typically receives compensation from one or more clients (the "Clients") for performing the analysis contained in these

reports (the “Reports”). The Reports may be distributed through other means, including via ConsenSys publications and other distributions.

The Reports are not an endorsement or indictment of any particular project or team, and the Reports do not guarantee the security of any particular project. This Report does not consider, and should not be interpreted as considering or having any bearing on, the potential economics of a token, token sale or any other product, service or other asset. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. No Report provides any warranty or representation to any Third-Party in any respect, including regarding the bugfree nature of code, the business model or proprietors of any such business model, and the legal compliance of any such business. No third party should rely on the Reports in any way, including for the purpose of making any decisions to buy or sell any token, product, service or other asset. Specifically, for the avoidance of doubt, this Report does not constitute investment advice, is not intended to be relied upon as investment advice, is not an endorsement of this project or team, and it is not a guarantee as to the absolute security of the project. CD owes no duty to any Third-Party by virtue of publishing these Reports.

PURPOSE OF REPORTS The Reports and the analysis described therein are created solely for Clients and published with their consent. The scope of our review is limited to a review of Solidity code and only the Solidity code we note as being within the scope of our review within this report. The Solidity language itself remains under development and is subject to unknown risks and flaws. The review does not extend to the compiler layer, or any other areas beyond Solidity that could present security risks. Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty.

CD makes the Reports available to parties other than the Clients (i.e., “third parties”) – on its website. CD hopes that by making these analyses publicly available, it can help the blockchain ecosystem develop technical best practices in this rapidly evolving area of innovation.

LINKS TO OTHER WEB SITES FROM THIS WEB SITE You may, through hypertext or other computer links, gain access to web sites operated by persons other than ConsenSys and CD. Such hyperlinks are provided for your reference and convenience only, and are the exclusive responsibility of such web sites’ owners. You agree that ConsenSys and CD are not responsible for



the content or operation of such Web sites, and that ConsenSys and CD shall have no liability to you or any other person or entity for the use of third party Web sites. Except as described below, a hyperlink from this web Site to another web site does not imply or mean that ConsenSys and CD endorses the content on that Web site or the operator or operations of that site. You are solely responsible for determining the extent to which you may use any content at any other web sites to which you link from the Reports. ConsenSys and CD assumes no responsibility for the use of third party software on the Web Site and shall have no liability whatsoever to any person or entity for the accuracy or completeness of any outcome generated by such software.

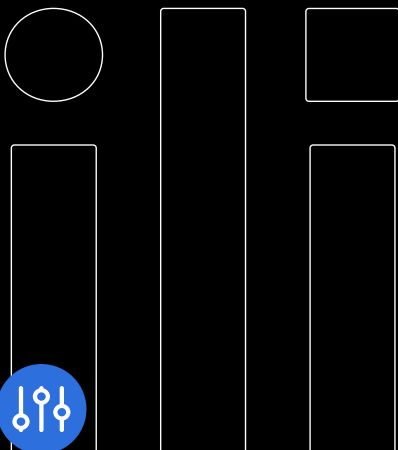
TIMELINESS OF CONTENT The content contained in the Reports is current as of the date appearing on the Report and is subject to change without notice. Unless indicated otherwise, by ConsenSys and CD.



Request a Security Review Today

Get in touch with our team to request a quote for a smart contract audit.

[CONTACT US](#)



AUDITS

FUZZING

SCRIBBLE

BLOG

TOOLS

RESEARCH

ABOUT

CONTACT

Subscribe to Our Newsletter

Stay up-to-date on our latest offerings, tools, and the world of blockchain security.

CONTACT

CAREERS


PRIVACY
POLICY

Email*

e-mail address

→

POWERED BY



CONSENSYS

