TECHFUND Audit Score
Smart contract Audit Report

NST Multi sign wallet Audit Report

Completed on 2022-06-09

Score **POSITIVE**

Risk level		
	Critical	0
	High	0
	Medium	0
	Low	0
	Note	2

Risk level detail

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
	Likelihood			

The tester arrives at the likelihood and impact estimates, they can now combine them to get a final severity rating for this risk. Note that if they have good business impact information, they should use that instead of the technical impact information.

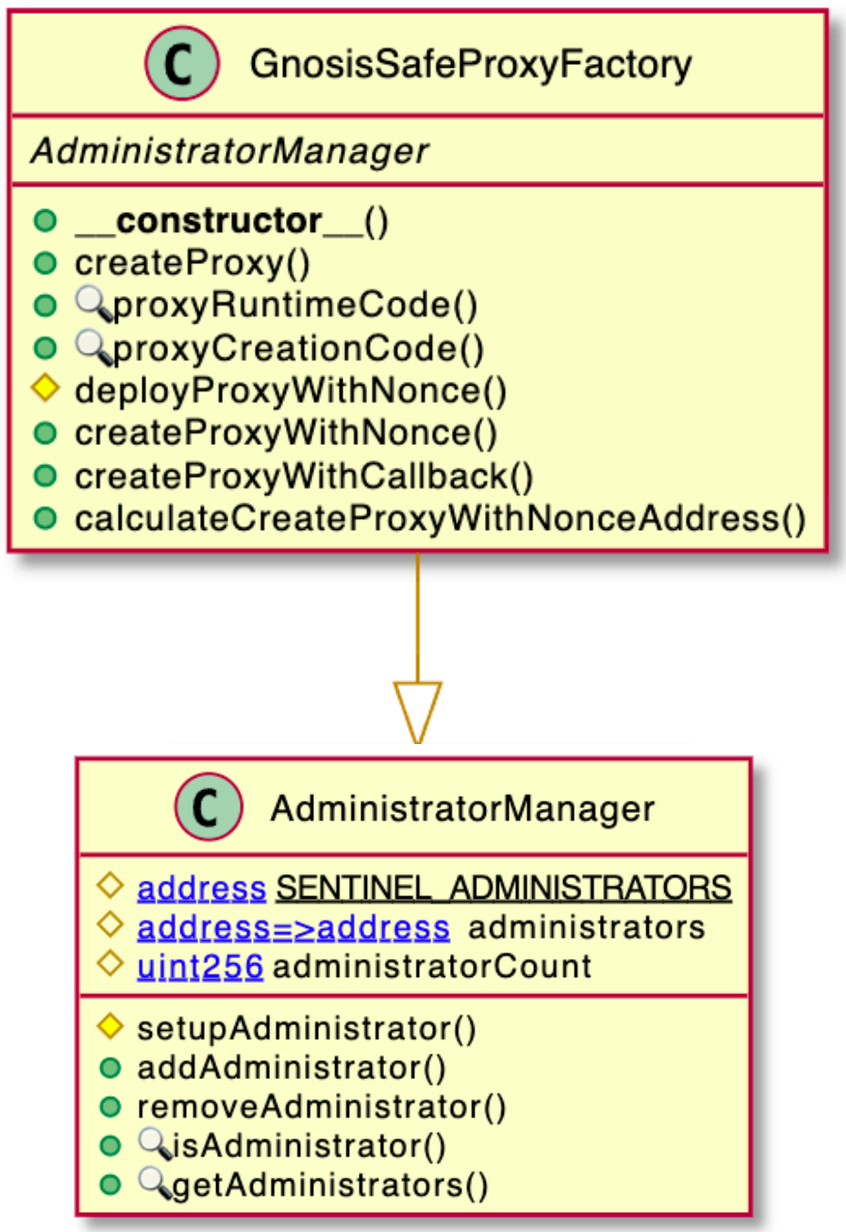
https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

Vulnerability Review

Number of warnings

Compiler Version	0
State Variable Default Visibility	0
Making state discoverable at fixed cost	1
Gas cost optimisation by batching	1
Integer Overflow / Underflow	0
Parity Multisig Bug	0
Callstack Depth Attack	0
Production Node modules security	0
Development Node modules security	0
Re-Entrancy	0
Double Withdrawal	0

Call Graph


GnosisSafeProxyFactory
AdministratorManager
__constructor__()
createProxy()
proxyRuntimeCode()
proxyCreationCode()
deployProxyWithNonce()
createProxyWithNonce()
createProxyWithCallback()
calculateCreateProxyWithNonceAddress()
AdministratorManager
address SENTINEL_ADMINISTRATORS
address=>address administrators
uint256 administratorCount
setupAdministrator()
addAdministrator()
removeAdministrator()
isAdministrator()
getAdministrators()

Making state discoverable at fixed cost

1

Ideally “state” should be discoverable at a fixed cost per transaction. Since administrator size can be anything, currently the cost of transaction can be potentially large. Iteration should be a client-side activity based on “total count” information. Provided the individual transactions always complete at a reasonable cost, a client can iterate endlessly.
Hence it is suggested to have a “totalAdministrators” and “getAdministrator(index)” instead of a single potentially large array by “getAdministrators”.
Since it is not a critical issue and is a user experience issue, it is mentioned under note.

```
function getAdministrators() public view returns (address[] memory) {
    address[] memory array = new address[](administratorCount);

    uint256 index = 0;
    address currentAdministrator = administrators[SENTINEL_ADMINISTRATORS];
    while (currentAdministrator != SENTINEL_ADMINISTRATORS) {
        array[index] = currentAdministrator;
        currentAdministrator = administrators[currentAdministrator];
        index++;
    }
    return array;
}
```

Gas cost optimisation by batching

1

Some of the calls to the functions are “overheads” for example checking “is administrator” and so on. Such overheads can be reduced by batching. For example in “addAdministrator” function, if we allow batch add we can save a lot of Gas. It is suggested to implement adding admins in batches to save Gas.

```
35  function addAdministrators(address[] calldata  administrator) public onlyAdministrator {
36      for( uint i=0;i < administrator.length ; i++ ){
37          require(administrator[i] != address(0) && administrator[i] != SENTINEL_ADMINISTRATORS, "GS10002");
38          require(administrators[administrator[i]] == address(0), "GS10003");
39          administrators[administrator[i]] = administrators[SENTINEL_ADMINISTRATORS];
40          administrators[SENTINEL_ADMINISTRATORS] = administrator[i];
41          administratorCount++;
42          emit AddedAdministrator(administrator[i]);
43      }
44  }
45
46
    address[] administrator   8 reference(s)      or(address administrator) public onlyAdministrator {
                                                  ount - 1 > 0  "GS10004");
    0   listen on all transactions   Search with transaction hash or address

    from        0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
    to          AdministratorManager.addAdministrators(address[]) 0xD06310d8dbF5ffC4E650f344f145a73048E2D96F
    gas         101957 gas
    transaction cost   88658 gas
    execution cost     88658 gas
```

```
gas                 67494 gas
transaction cost    58690 gas
execution cost      58690 gas
input               0xc99...35cb2
decoded input       {
                        "address administrator":
                    }
decoded output      {}
```

Adding 2 admins
Current Method : Gas 134988
Batch Method : Gas 101957