TECHFUND Audit Score
Smart contract Audit Report

NST Multi sign wallet Audit Report

Completed on 2022-05-12
#361f0375424085e3edb87b736c65e7412fc1eb29

Score **POSITIVE**

Risk level		
	Critical	0
	High	0
	Medium	0
	Low	1
	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.

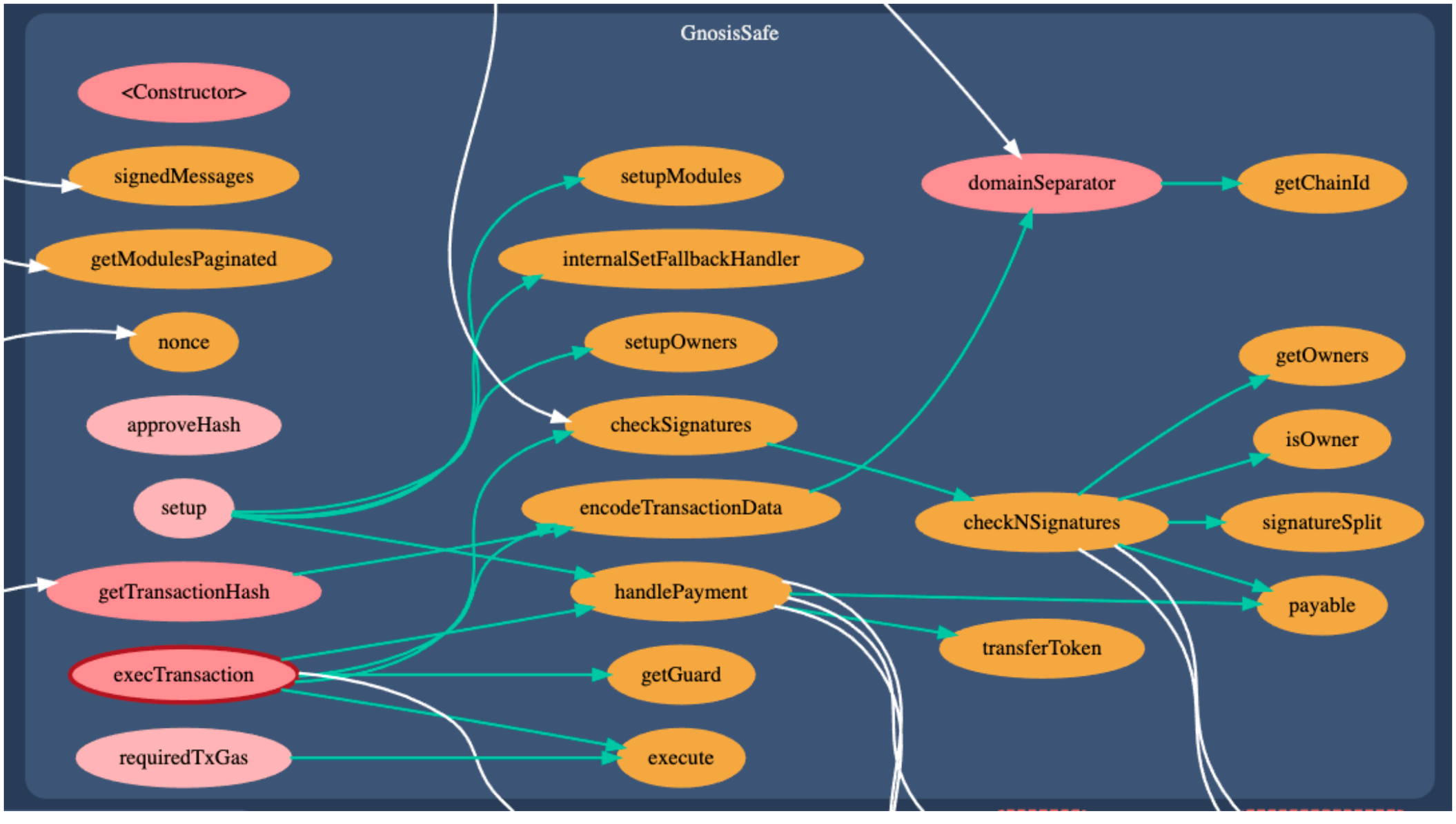
https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

Vulnerability Review

Number of warnings

Compiler Version	0
State Variable Default Visibility	0
Arithmetic Operations	0
Inherited / Shadowed variable name	1
Integer Overflow / Underflow	0
Parity Multisig Bug	0
Callstack Depth Attack	0
Deployment improvements	1
Deployment : Contact Code size	1
Re-Entrancy	0
Double Withdrawal	0

Call Graph : GnosisSafe



Inherited / Shadowed variable name 1

Solidity allows for ambiguous naming of state variables when inheritance is used. Contract `A` with a variable `x` could inherit contract `B` that also has a state variable `x` defined. This would result in two separate versions of `x`, one of them being accessed from contract `A` and the other one from contract `B`. In more complex contract systems this condition could go unnoticed and subsequently lead to security issues.

Currently it is not an issue but GnosisSafe (Inherits : OwnerManager) has defined “owners” inside OwnerManager and is again being defined inside GnosisSafe. We suggest to change to another variable name to prevent any future issues.

```
contract GnosisSafe is
    EtherPaymentFallback,
    Singleton,
    ModuleManager,
    OwnerManager,
    SignatureDecoder,
    SecuredTokenTransfer,
    ISignatureValidatorConstants,
    FallbackManager,
    StorageAccessible,
    GuardManager
{
```

```
contract OwnerManager is SelfAuthorized {
    event AddedOwner(address owner);
    event RemovedOwner(address owner);
    event ChangedThreshold(uint256 threshold);

    address internal constant SENTINEL_OWNERS = address
    (0x1);

    mapping(address => address) internal owners;
    uint256 internal ownerCount;
    uint256 internal threshold;
```

```
address[] memory owners = getOwners();
currentOwnerGroup = SENTINEL_OWNERS;
for (j = 0; j < owners.length; j++) {
    address owner = owners[j];
    if (owner == SENTINEL_OWNERS) {
        continue;
    }
```

Deployment improvements 1

Instead of supplying the salt parameter directly to the CREATE2 operation, we suggest to first hash it together with the caller of the deploy function. This means that only the original user will be able to call into this function and deploy into the address they had parked. Any different msg.sender would yield a different salt, and thus a different deployment address.

bytes32 newsalt = keccak256(abi.encodePacked(salt, msg.sender));

```
ftrace | funcSig
function create(
    bytes memory code,
    uint256 salt,
    bytes calldata data
) external returns (address) {
    address addr;

    // solhint-disable-next-line no-inline-assembly
    assembly {
        addr := create2(0, add(code, 0x20), mload
        (code), salt)
        if iszero(extcodesize(addr)) {
            revert(0, 0)
        }
    }
```

TECHFUND Audit Score

Smart contract Audit Report

Deployment : Contact Code size

1

On November 22, 2016 the Spurious Dragon hard-fork introduced EIP-170 which added a smart contract size limit of 24.576 kb. This limit was introduced to prevent denial-of-service (DOS) attacks. Any call to a contract is relatively cheap gas-wise. However, the impact of a contract call for Ethereum nodes increases disproportionately depending on the called contract code's size (reading the code from disk, pre-processing the code, adding data to the Merkle proof). Whenever you have such a situation where the attacker requires few resources to cause a lot of work for others, you get the potential for DOS attacks.

https://github.com/ethereum/EIPs/blob/master/EIPS/eip-170.md

Hardhat provides optimiser settings to handle such cases and we suggest to set it enabled as shown in screenshot.

```
Warning: Contract code size is 25138 bytes and exceeds 24576 bytes (a limit introduced in Spurious Dragon). This contract may not be deployable on mainnet. Consider enabling the optimizer (with a low "runs" value!), turning off revert str
ings, or using libraries.
  --> contracts/GnosisSafe.sol:19:1:
   |
19 | contract GnosisSafe is
   | ^ (Relevant source part starts here and spans across multiple lines).
```

```
settings: {
  optimizer: {
    enabled: true,
    runs: 1000,
  },
},
```