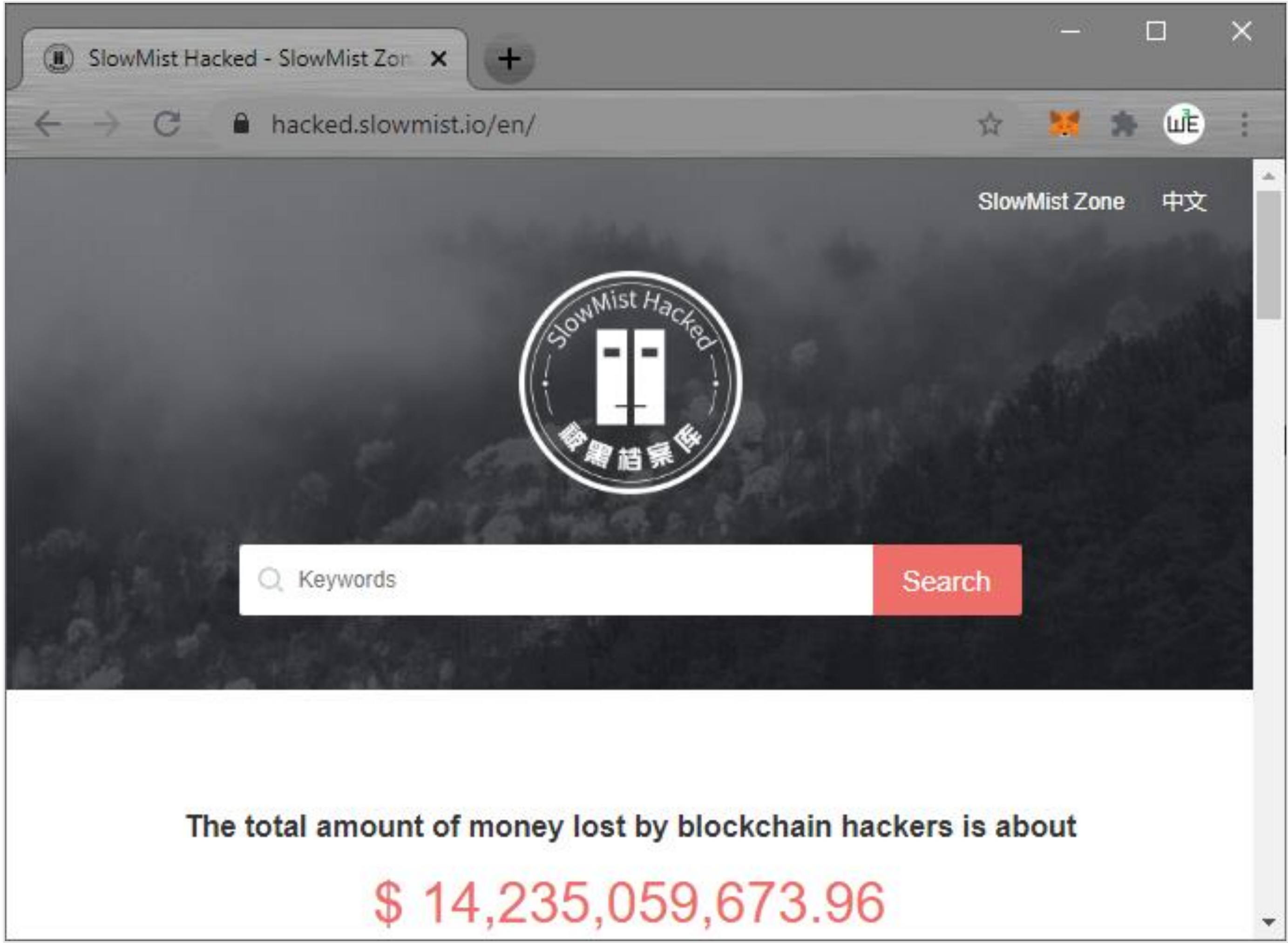
PD-12.0 Security

PD-12.1 Hacks and Weaknesses



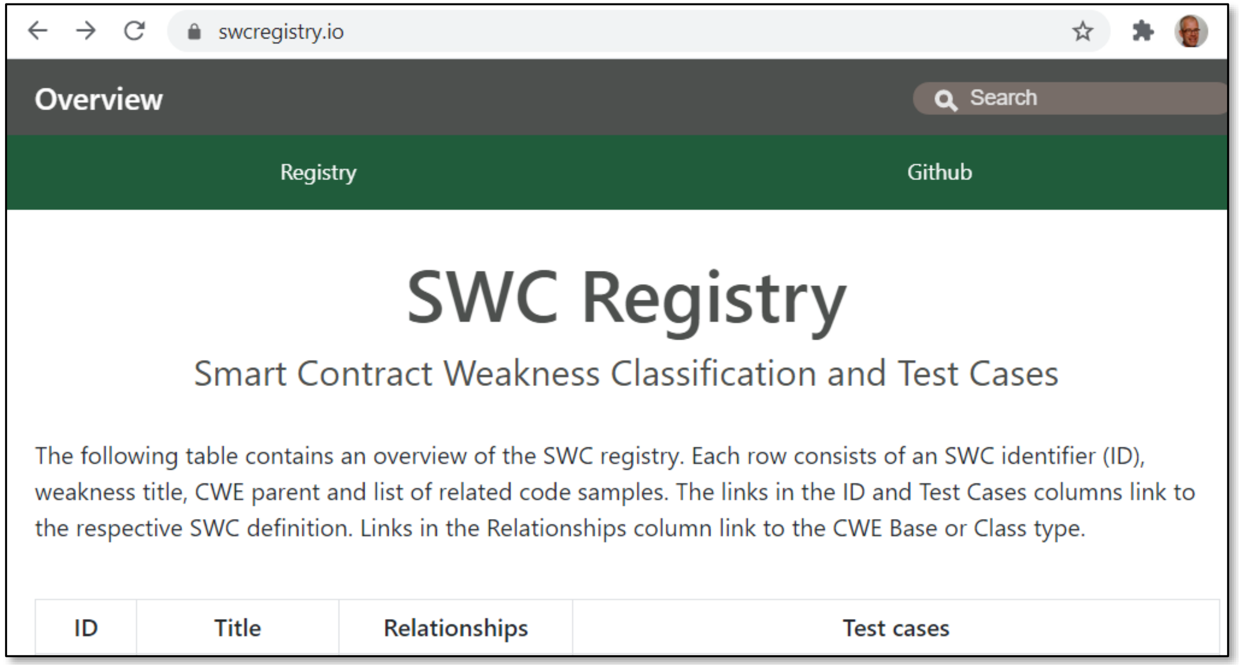
The total amount of money lost by blockchain hackers is about

$ 14,235,059,673.96

https://hacked.slowmist.io/en

PD-12.1 Smart Contract Weakness Classification Registry

SWC Registry



SWC Registry

Smart Contract Weakness Classification and Test Cases

The following table contains an overview of the SWC registry. Each row consists of an SWC identifier (ID), weakness title, CWE parent and list of related code samples. The links in the ID and Test Cases columns link to the respective SWC definition. Links in the Relationships column link to the CWE Base or Class type.

ID	Title	Relationships	Test cases

CWE Common Weakness Enumeration

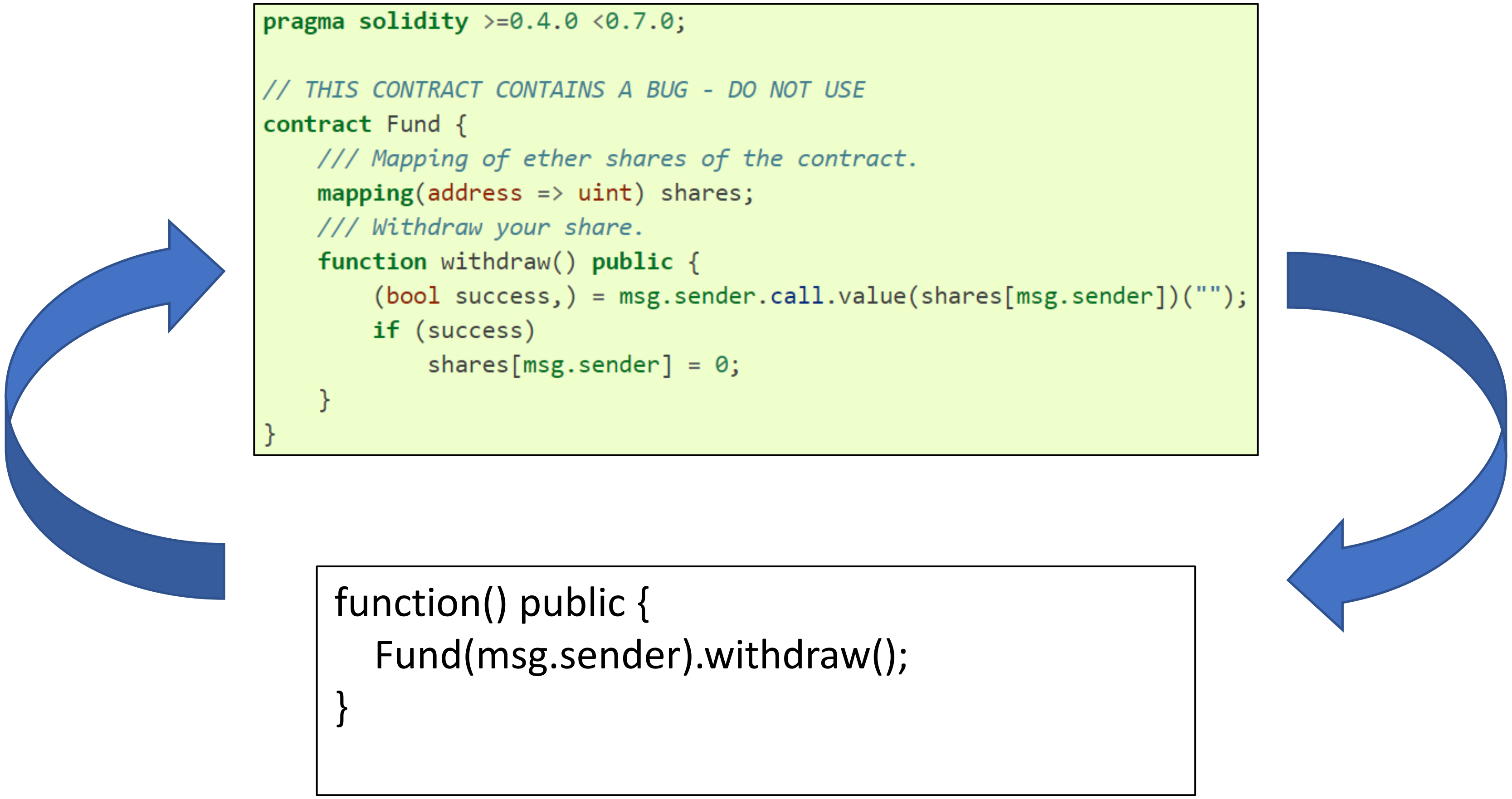
A Community-Developed List of Software Weakness Types

https://cwe.mitre.org/

https://swcregistry.io

https://github.com/SmartContractSecurity/SWC-registry

PD-12.2 Reentrancy attack

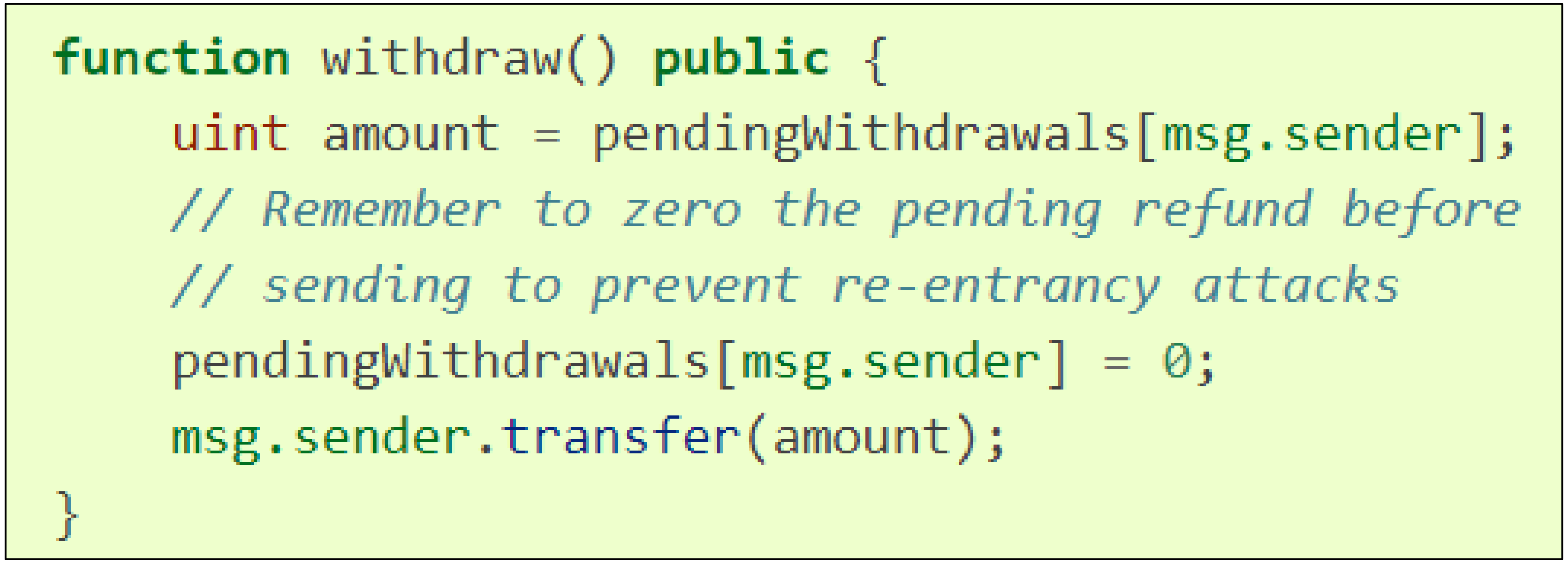


```
pragma solidity >=0.4.0 <0.7.0;

// THIS CONTRACT CONTAINS A BUG - DO NOT USE
contract Fund {
    /// Mapping of ether shares of the contract.
    mapping(address => uint) shares;
    /// Withdraw your share.
    function withdraw() public {
        (bool success,) = msg.sender.call.value(shares[msg.sender])("");
        if (success)
            shares[msg.sender] = 0;
    }
}
```

```
function() public {
   Fund(msg.sender).withdraw();
}
```

https://blog.openzeppelin.com/exploiting-uniswap-from-reentrancy-to-actual-profit/

PD-12.2 Use the Checks-Effects-Interactions

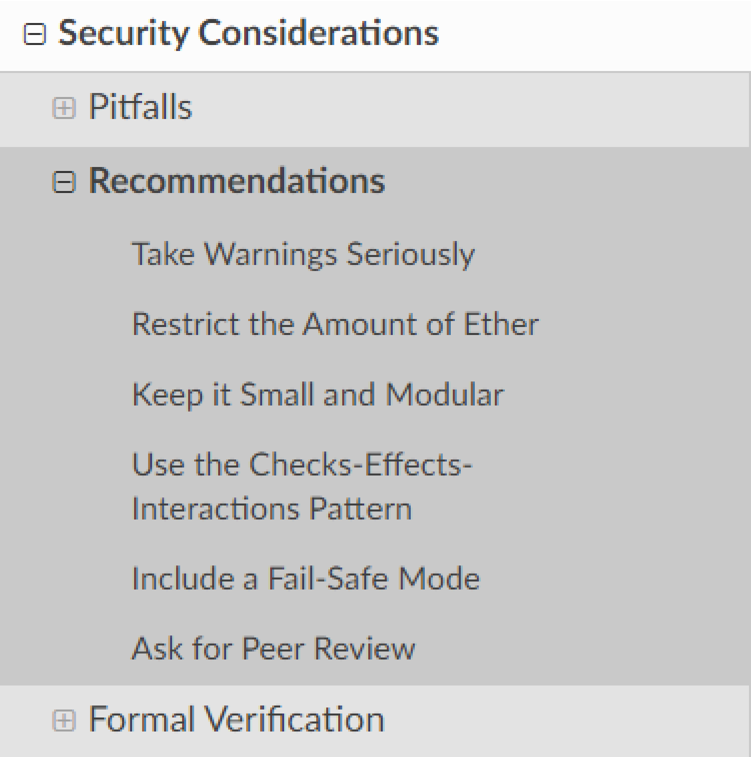


```
function withdraw() public {
    uint amount = pendingWithdrawals[msg.sender];
    // Remember to zero the pending refund before
    // sending to prevent re-entrancy attacks
    pendingWithdrawals[msg.sender] = 0;
    msg.sender.transfer(amount);
}
```

https://solidity.readthedocs.io/en/latest/security-considerations.html?highlight=check%20effects#re-entrancy

https://solidity.readthedocs.io/en/latest/security-considerations.html?highlight=check%20effects#use-the-checks-effects-interactions-pattern

PD-12.3 Security Best Practices: Solidity manual



- ⊟ **Security Considerations**
  - ⊞ Pitfalls
  - ⊟ **Recommendations**
    - Take Warnings Seriously
    - Restrict the Amount of Ether
    - Keep it Small and Modular
    - Use the Checks-Effects-Interactions Pattern
    - Include a Fail-Safe Mode
    - Ask for Peer Review
  - ⊞ Formal Verification

https://docs.soliditylang.org/en/latest/security-considerations.html

PD-12.3 Measures:
Smart Contract Security Verification Standard

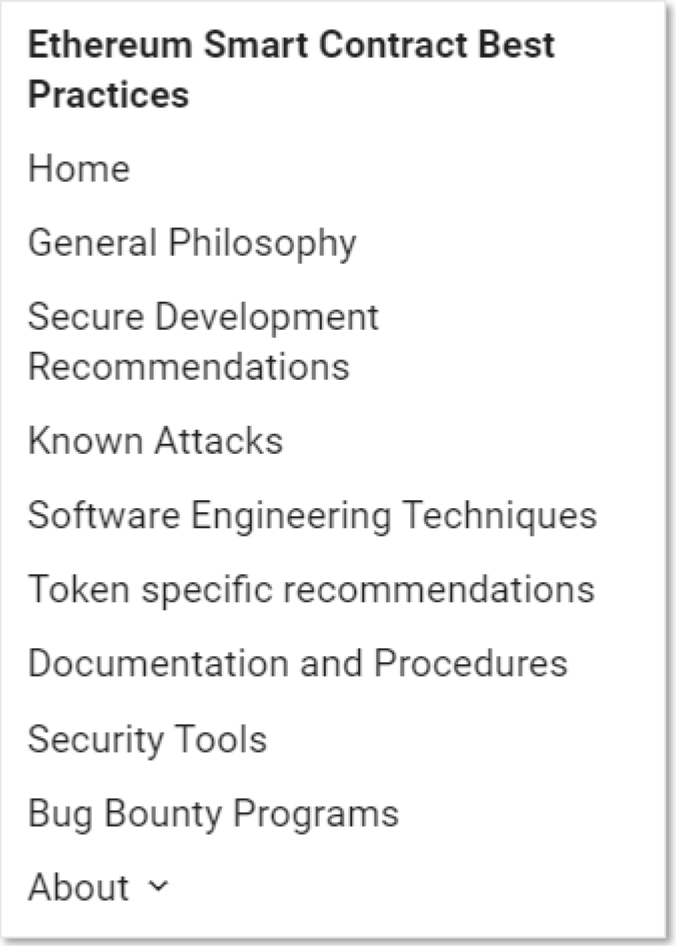
V1: Architecture, Design and Threat Modelling
V2: Access Control
V3: Blockchain Data
V4: Communications
V5: Arithmetic
V6: Malicious Input Handling
V7: Gas Usage & Limitations
V8: Business Logic
V9: Denial of Service
V10: Token
V11: Code Clarity
V12: Test Coverage
V13: Known Attacks

Based on:
OWASP

https://securing.github.io/SCSVS/

https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project

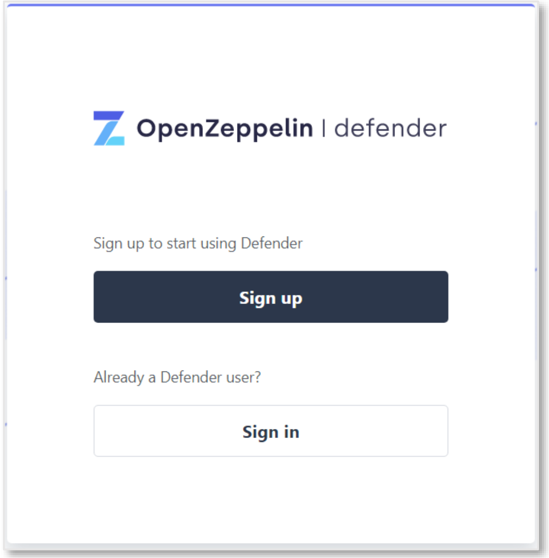
PD-12.3 Consensys security best practices


Ethereum Smart Contract Best Practices
Home
General Philosophy
Secure Development Recommendations
Known Attacks
Software Engineering Techniques
Token specific recommendations
Documentation and Procedures
Security Tools
Bug Bounty Programs
About

https://consensys.github.io/smart-contract-best-practices

PD-12.3 OpenZeppelin Defender Best practices

TITLE	CATEGORY	RATING	EFFORT
Test Contract Upgrades	Testing	CRITICAL	MEDIUM
Data Out of Sync	Monitoring	CRITICAL	MEDIUM
Privileged Administrator Transactions	Monitoring	CRITICAL	MEDIUM
Spikes in Account Activity	Monitoring	CRITICAL	LARGE
Implement Reentrancy Protections	Development	CRITICAL	SMALL
Recast Variables Safely	Development	CRITICAL	SMALL
Assert Revert Reasons	Testing	CRITICAL	SMALL
Drop In System Funds	Monitoring	CRITICAL	MEDIUM
Post-Mortem Analysis	Operations	CRITICAL	MEDIUM
Secure All Administrative Keys	Operations	CRITICAL	MEDIUM
Access Arrays Using Enumeration and Pagination	Development	CRITICAL	MEDIUM
Prevent Replay Attacks When Using Signatures	Development	CRITICAL	MEDIUM
Test Emission of Events	Testing	HIGH	SMALL
Collateral Ratios	Monitoring	HIGH	MEDIUM
Dependency Changes	Monitoring	HIGH	MEDIUM
Emergency Response Plan	Operations	HIGH	MEDIUM
Control Growth of Arrays	Development	HIGH	MEDIUM
Use the Offer-Accept Pattern for Transferring Admin Role	Development	HIGH	MEDIUM
Large Value Transactions	Monitoring	HIGH	MEDIUM
Avoid Packed Encoding When Hashing	Development	HIGH	SMALL
Do Not Use Solidity's `transfer` Function	Development	HIGH	SMALL
Achieve High Smart Contract Test Coverage	Testing	HIGH	LARGE
Use Low-Level Calls Carefully	Development	HIGH	MEDIUM
Use PullPayment When Sending ETH	Development	HIGH	SMALL
Asset Attacks and Issues	Monitoring	HIGH	SMALL
Significant Price Changes	Monitoring	HIGH	SMALL
Spikes in Failed Transactions	Monitoring	HIGH	LARGE
Emit Events on All State Changes	Development	HIGH	MEDIUM
Use Indexed Event Parameters	Development	HIGH	SMALL
Spikes in Function Calls	Monitoring	HIGH	MEDIUM
Avoid Implicit Function Arguments	Development	HIGH	SMALL
Spikes in Low Value Transactions	Monitoring	NORMAL	MEDIUM
Do Not Track Time With Block Numbers	Development	NORMAL	SMALL
Minimize Division Errors	Development	NORMAL	SMALL
Include Revert Reasons	Development	NORMAL	SMALL
Network Congestion	Monitoring	NORMAL	MEDIUM
Unused Tokens or Funds	Monitoring	NORMAL	MEDIUM
Expiring Assets	Monitoring	NORMAL	SMALL
Declare Constants Explicitly	Development	NORMAL	SMALL



Sign up to start using Defender

Already a Defender user?

https://defender.openzeppelin.com/#/advisor/doclist

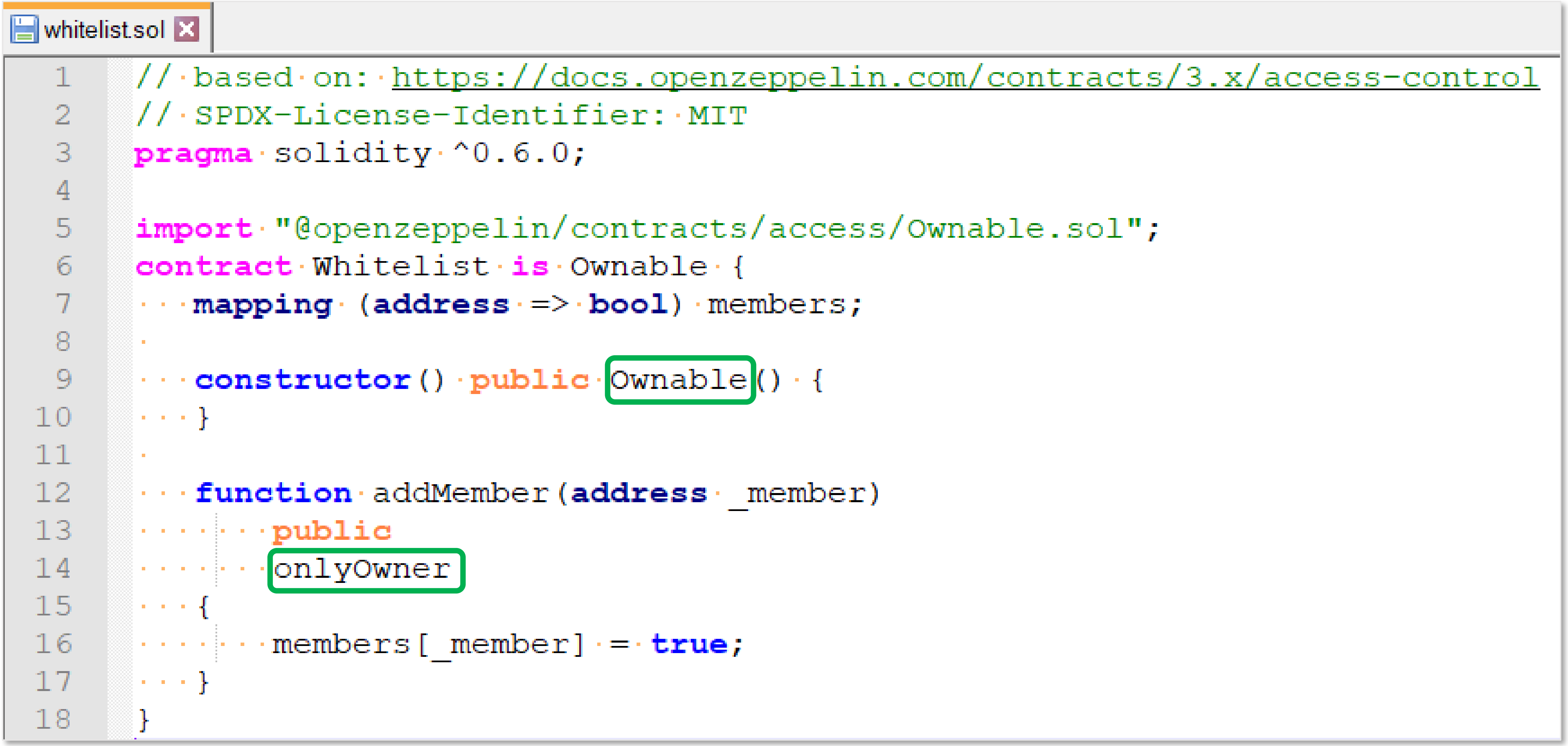
PD-12.3 Token checklist

Token	Feature	Known Vulnerabilities
ERC20	Allowance	Double withdrawal (front-running)
		Not accounting for the tokens that try to prevent multiple withdrawal attack
		Unprotected transferFrom()
	External Calls	Unchecked Call Return Value
		DoS with unexpected revert
	Transfers	Might return False instead of Revert
		Missing return value
	BalanceOf()	Internal Accounting discrepancy with the Actual Balance
	Blacklistable	Blacklisted addresses cannot receive or send tokens
	Mintable / Burnable	TotalSupply can change by trusted actors
	Pausable	All functionalities can be paused by trusted actors
Deflationary Tokens	Take fees from transfers	Internal Accounting discrepancy with the Actual Balance
Inflationary Tokens	AirDrop interest to token holders	Internal Accounting discrepancy with the Actual Balance
ERC1400	Permissioned Addresses	Can block transfers from/to specific addresses
	Forced Transfers	Trusted actors have the ability to transfer funds however they choose
ERC777	Callbacks / Hooks	Reentrancy
		Receiver mining GasToken
		Receiver blocks the transfer
ERC1644	Forced Transfers	Controller has the ability to steal funds
ERC621	Control of totalSupply	totalSupply can be changed by trusted actors
ERC884	Cancel and Reissue	Token implementers have the ability to cancel an address and move its tokens to a new address
	Whitelisting	Tokens can only be sent to whitelisted addresses

https://consensys.net/diligence/blog/2020/11/token-interaction-checklist

https://gist.github.com/shayanb/cd495e23c7cf1a8b269f8ce7fd198538

PD-12.4 Access control 1/2: Ownable



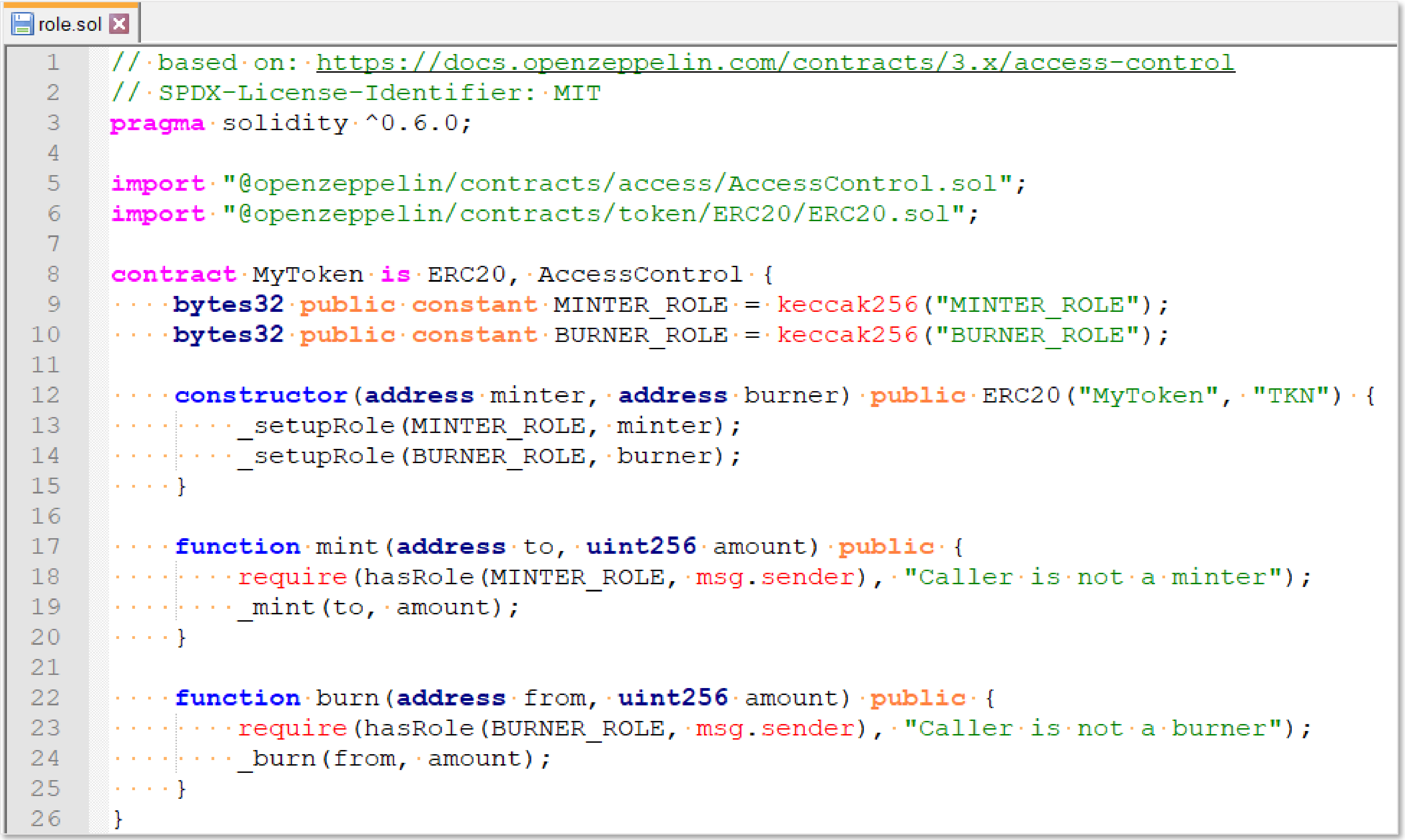
```
// based on: https://docs.openzeppelin.com/contracts/3.x/access-control
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;

import "@openzeppelin/contracts/access/Ownable.sol";
contract Whitelist is Ownable {
    mapping (address => bool) members;

    constructor() public Ownable() {
    }

    function addMember(address _member)
        public
        onlyOwner
    {
        members[_member] = true;
    }
}
```

https://medium.com/coinmonks/guide-to-ownership-and-access-control-in-solidity-f2d99f63c6d4

https://docs.openzeppelin.com/contracts/3.x/access-control

https://github.com/web3examples/ethereum/blob/master/security_examples/accesscontrol/contracts/whitelist.sol

PD-12.4 Access control 2/2: Roles



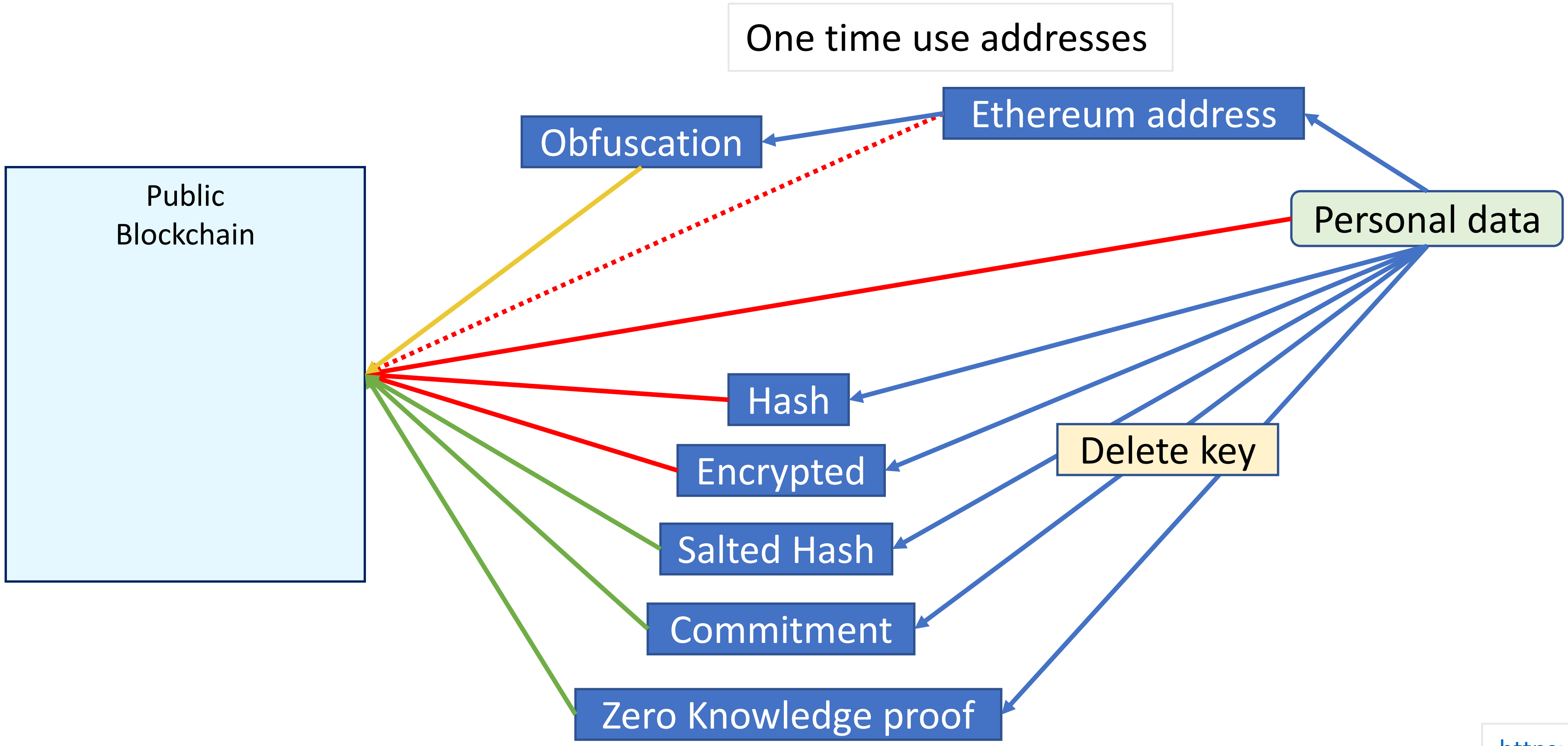
role.sol

```
// based on: https://docs.openzeppelin.com/contracts/3.x/access-control
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;

import "@openzeppelin/contracts/access/AccessControl.sol";
import "@openzeppelin/contracts/token/ERC20/ERC20.sol";

contract MyToken is ERC20, AccessControl {
    bytes32 public constant MINTER_ROLE = keccak256("MINTER_ROLE");
    bytes32 public constant BURNER_ROLE = keccak256("BURNER_ROLE");

    constructor(address minter, address burner) public ERC20("MyToken", "TKN") {
        _setupRole(MINTER_ROLE, minter);
        _setupRole(BURNER_ROLE, burner);
    }

    function mint(address to, uint256 amount) public {
        require(hasRole(MINTER_ROLE, msg.sender), "Caller is not a minter");
        _mint(to, amount);
    }

    function burn(address from, uint256 amount) public {
        require(hasRole(BURNER_ROLE, msg.sender), "Caller is not a burner");
        _burn(from, amount);
    }
}
```

https://docs.openzeppelin.com/contracts/3.x/access-control

https://github.com/web3examples/ethereum/blob/master/security_examples/accesscontrol/contracts/role.sol

(GDPR)



encryption = pseudonymization
≠ anonymization

https://en.wikipedia.org/wiki/Commitment_scheme

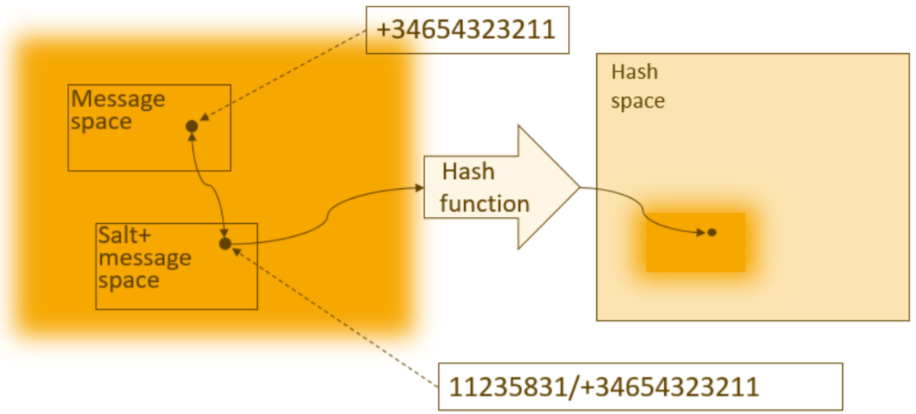
https://www.cnil.fr/sites/default/files/atoms/files/blockchain.pdf

https://www.cravath.com/files/Uploads/Documents/Publications/3900063_1.pdf

https://www.eublockchainforum.eu/sites/default/files/reports/20181016_report_gdpr.pdf

https://www.europarl.europa.eu/RegData/etudes/STUD/2019/634445/EPRS_STU(2019)634445_EN.pdf

PD-12.5 GDPR: Salted hash



	situation a) hash is used to replace a unique attribute in a dataset	situation b) hash is used as a one-time value to notarise the state of a dataset
reversal risk (reverse engineering)	medium. brute force can be considered viable if the size of the input is known or within a small range (e.g. ssn, password, name) can potentially be mitigated using a salt or pepper.	low. reverse engineering is non-trivial as the size of the input can range from a few bytes to hundreds of terabytes and be coupled with multiple layers of hashing.
linkability risk (via data analysis)	high. it is possible to conduct pattern analysis and trace data back to the individual, potentially with the help of other sources of information.	low. each hash is unique. there is no obvious way to cross-analyse the data.

https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf#page=20

https://www.eublockchainforum.eu/sites/default/files/reports/20181016_report_gdpr.pdf#page=22

https://edps.europa.eu/sites/edp/files/publication/19-10-30_aepd-edps_paper_hash_final_en.pdf#page=18

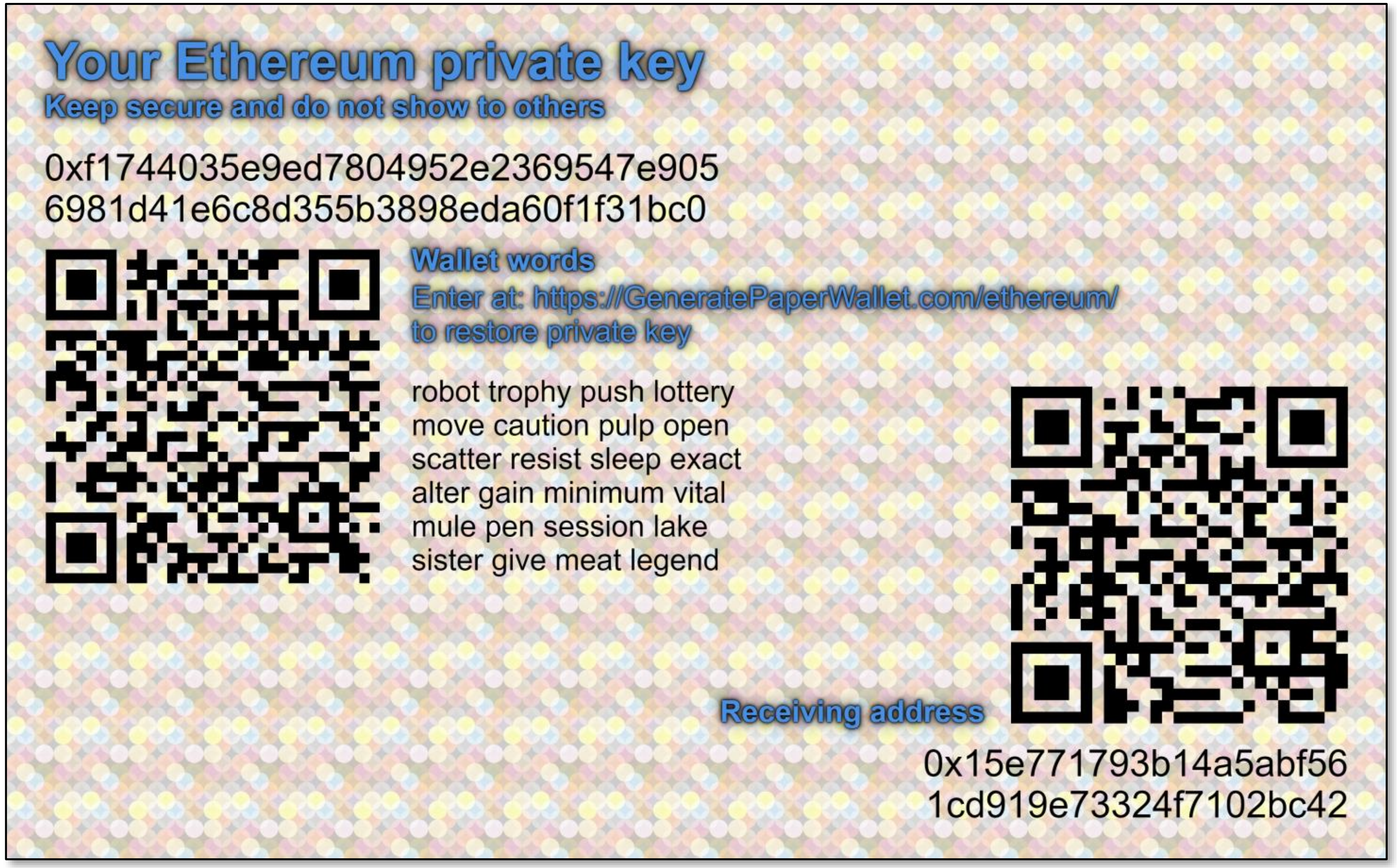
PD-12.6 Use a password manager

To store:

- Pincodes
- Pass phrases
- Addresses & private keys

https://keepass.info/

PD-12.5 Paper wallet



Your Ethereum private key

Keep secure and do not show to others

0xf1744035e9ed7804952e2369547e905
6981d41e6c8d355b3898eda60f1f31bc0

Wallet words

Enter at: https://GeneratePaperWallet.com/ethereum/
to restore private key

robot trophy push lottery
move caution pulp open
scatter resist sleep exact
alter gain minimum vital
mule pen session lake
sister give meat legend

Receiving address

0x15e771793b14a5abf56
1cd919e73324f7102bc42

Risks:

- Online creation not save
  - Not sufficiently random
  - Eaves dropped
- Lost
- Stolen
- Fire / Water

https://github.com/hubernautmartin/generator

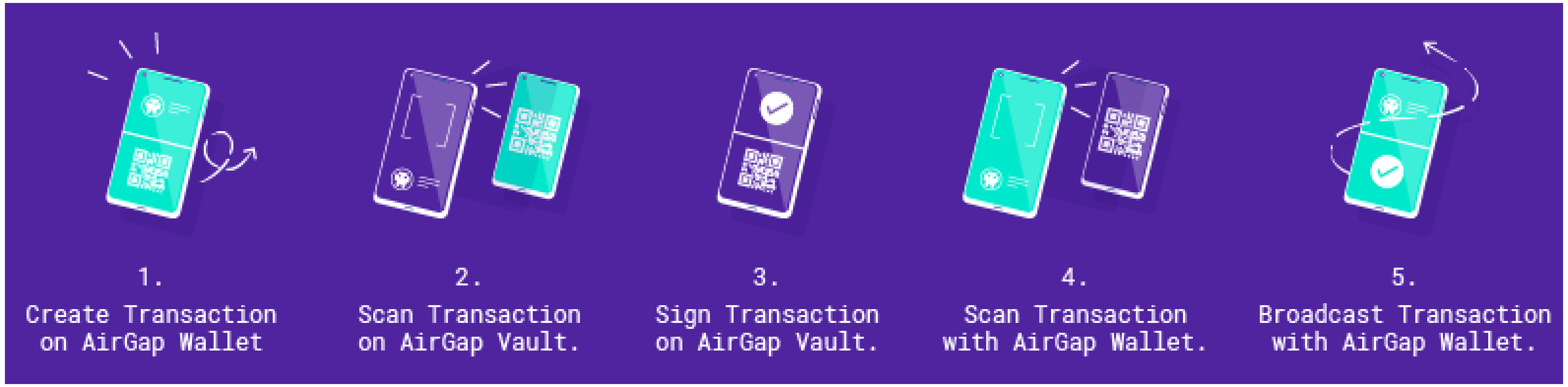
PD-12.6 Steel wallet

https://cryptosteel.com

https://www.blockplate.com/

PD-12.6 Offline / airgap

- Offline signing



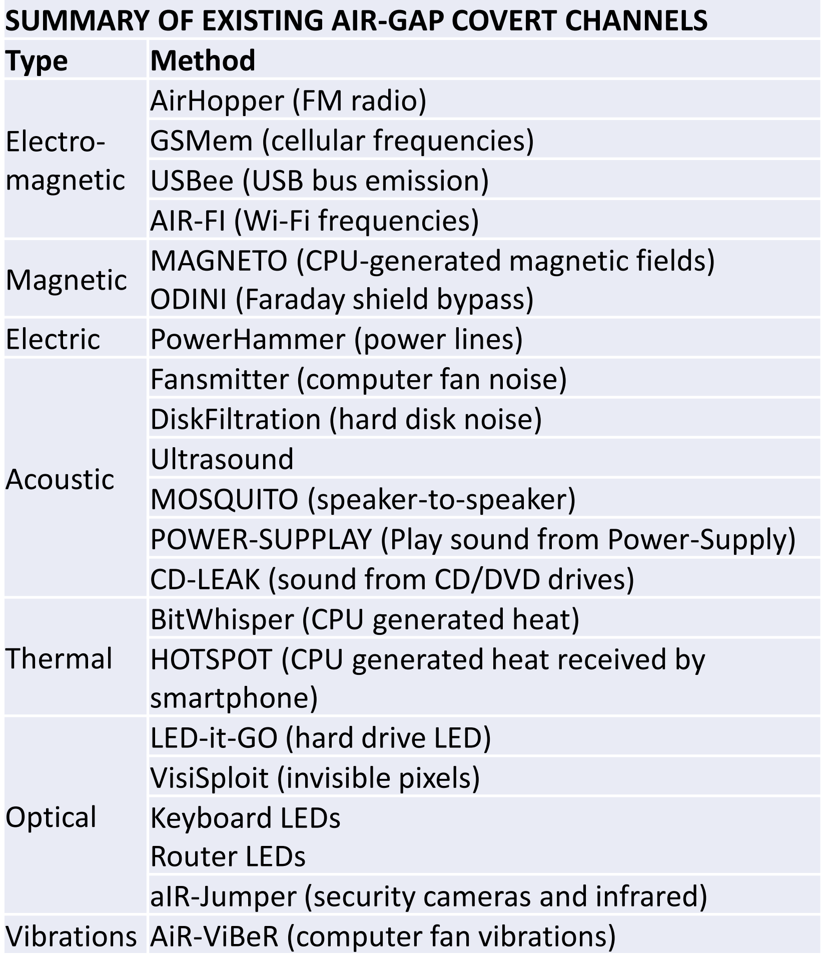
1. Create Transaction on AirGap Wallet

2. Scan Transaction on AirGap Vault.

3. Sign Transaction on AirGap Vault.

4. Scan Transaction with AirGap Wallet.

5. Broadcast Transaction with AirGap Wallet.



SUMMARY OF EXISTING AIR-GAP COVERT CHANNELS

Type	Method
Electro-magnetic	AirHopper (FM radio)
	GSMem (cellular frequencies)
	USBee (USB bus emission)
	AIR-FI (Wi-Fi frequencies)
Magnetic	MAGNETO (CPU-generated magnetic fields)
	ODINI (Faraday shield bypass)
Electric	PowerHammer (power lines)
Acoustic	Fansmitter (computer fan noise)
	DiskFiltration (hard disk noise)
	Ultrasound
	MOSQUITO (speaker-to-speaker)
	POWER-SUPPLAY (Play sound from Power-Supply)
	CD-LEAK (sound from CD/DVD drives)
Thermal	BitWhisper (CPU generated heat)
	HOTSPOT (CPU generated heat received by smartphone)
Optical	LED-it-GO (hard drive LED)
	VisiSploit (invisible pixels)
	Keyboard LEDs
	Router LEDs
	aIR-Jumper (security cameras and infrared)
Vibrations	AiR-ViBeR (computer fan vibrations)

https://airgap.it/

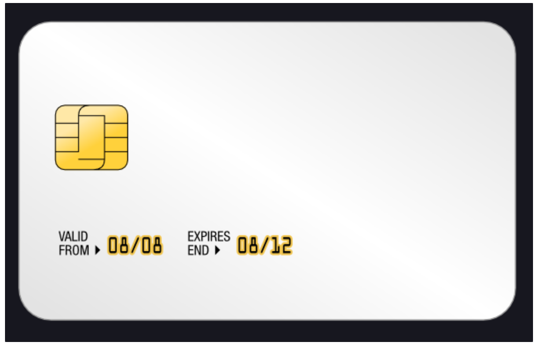
https://medium.com/airgap-it/airgap-the-step-by-step-guide-c4c3d3fe9a05

https://threatpost.com/air-gap-attack-turns-memory-wifi/162358

https://arxiv.org/pdf/2012.06884.pdf

PD-12.6 HSM & Smartcards

- Physical security
- Logical security key servers



https://i.blackhat.com/USA-19/Thursday/us-19-Campana-Everybody-Be-Cool-This-Is-A-Robbery.pdf

https://www.unboundtech.com/how-to-hack-an-hardware-security-module/

https://loomx.io/developers/docs/en/hsm.html

https://www.yubico.com/products/hardware-security-module/

https://ethereum.stackexchange.com/questions/73192/using-aws-cloudhsm-to-sign-transactions

PD-12.6 Hardware wallets

https://shop.ledger.com/products/ledger-nano-s

https://shop.ledger.com/pages/ledger-nano-x

Risks:

- Recovery phrase
- Pin code
- Private keys can be retrieved
- Physical access
- Firmware updates
- Hacks

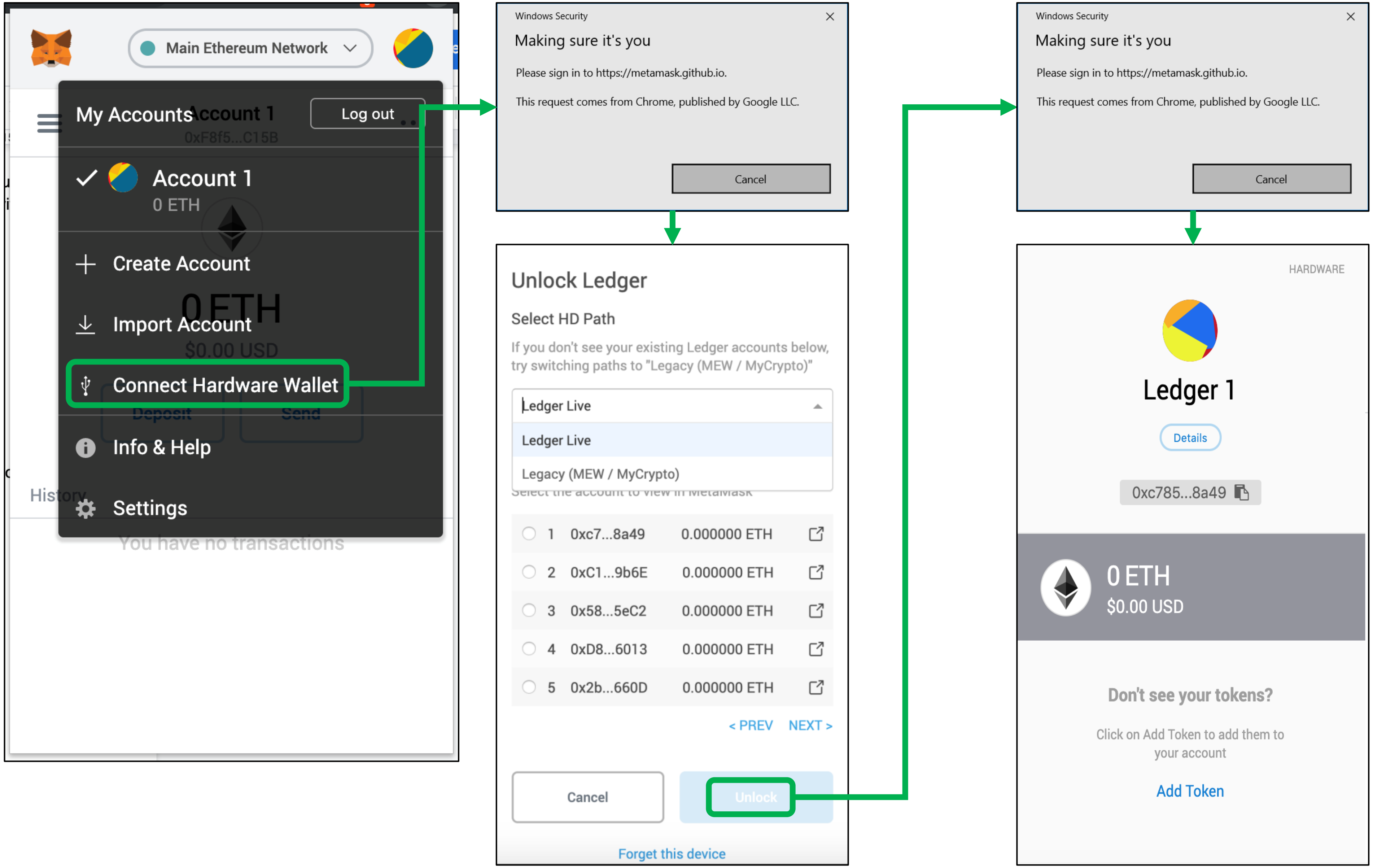
https://wallet.fail/wallets/nanos/

https://saleemrashid.com/2018/03/20/breaking-ledger-security-model/

https://ledger.readthedocs.io/en/latest/

https://www.ledger.com/

PD-12.7 Connect Ledger to MetaMask



Main Ethereum Network

My Accounts

Log out

0xF8f5...C15B

✓ Account 1
0 ETH

＋ Create Account

Import Account

0 ETH
$0.00 USD

Connect Hardware Wallet

Deposit Send

Info & Help

History

Settings

You have no transactions

Windows Security

Making sure it's you

Please sign in to https://metamask.github.io.

This request comes from Chrome, published by Google LLC.

Cancel

Unlock Ledger

Select HD Path

If you don't see your existing Ledger accounts below, try switching paths to "Legacy (MEW / MyCrypto)"

Ledger Live
- Ledger Live
- Legacy (MEW / MyCrypto)

Select the account to view in MetaMask

○ 1	0xc7...8a49	0.000000 ETH	↗
○ 2	0xC1...9b6E	0.000000 ETH	↗
○ 3	0x58...5eC2	0.000000 ETH	↗
○ 4	0xD8...6013	0.000000 ETH	↗
○ 5	0x2b...660D	0.000000 ETH	↗

< PREV NEXT >

Cancel Unlock

Forget this device

Windows Security

Making sure it's you

Please sign in to https://metamask.github.io.

This request comes from Chrome, published by Google LLC.

Cancel

HARDWARE

Ledger 1

Details

0xc785...8a49

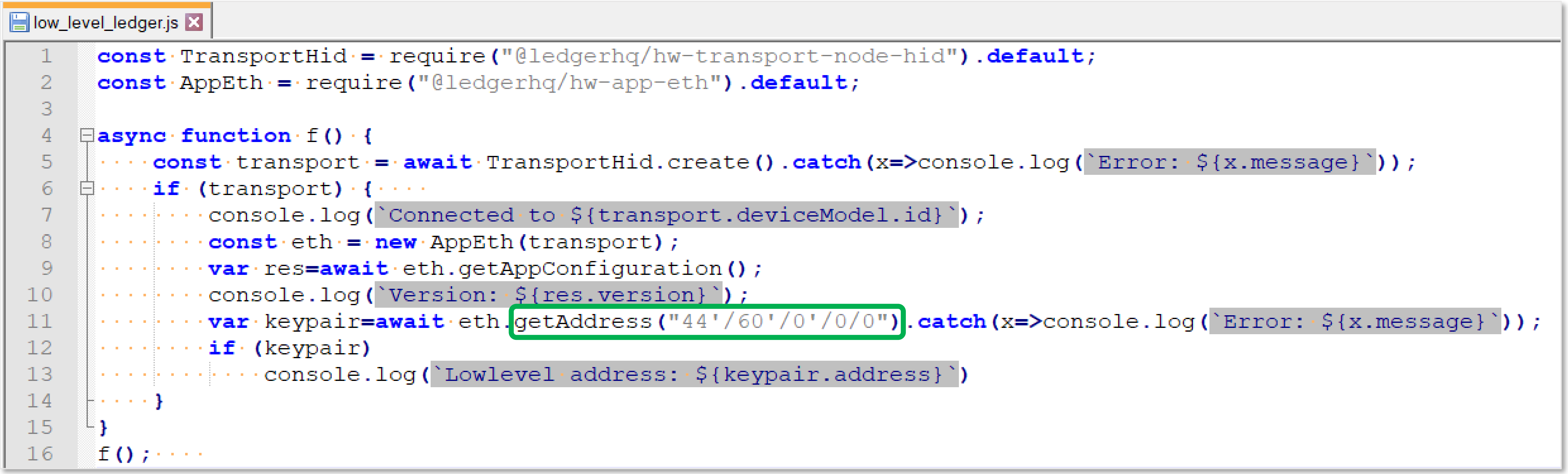
0 ETH
$0.00 USD

Don't see your tokens?

Click on Add Token to add them to your account

Add Token

PD-12.7 Ledger via Javascript (low level API)

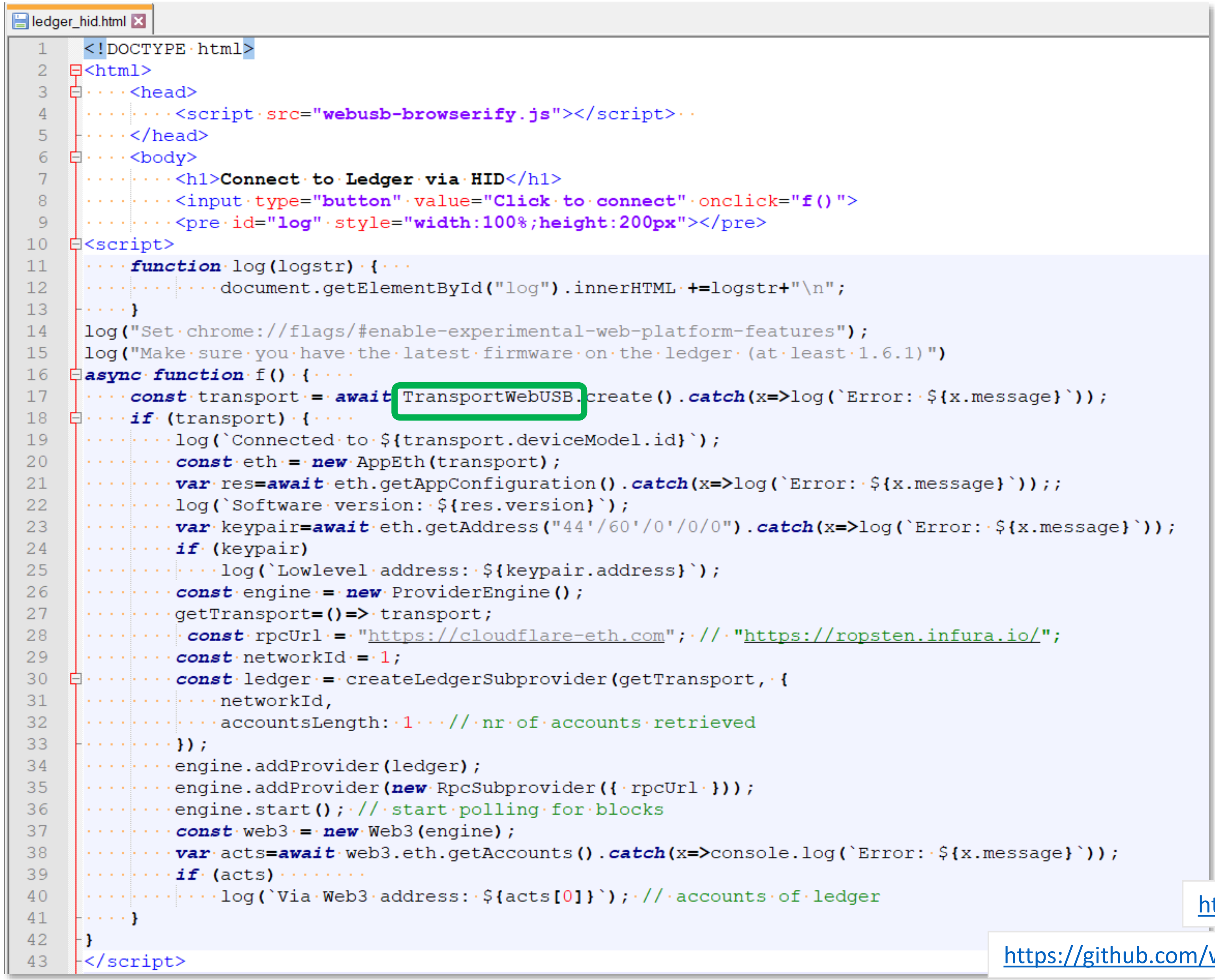


```
const TransportHid = require("@ledgerhq/hw-transport-node-hid").default;
const AppEth = require("@ledgerhq/hw-app-eth").default;

async function f() {
    const transport = await TransportHid.create().catch(x=>console.log(`Error: ${x.message}`));
    if (transport) {
        console.log(`Connected to ${transport.deviceModel.id}`);
        const eth = new AppEth(transport);
        var res=await eth.getAppConfiguration();
        console.log(`Version: ${res.version}`);
        var keypair=await eth.getAddress("44'/60'/0'/0/0").catch(x=>console.log(`Error: ${x.message}`));
        if (keypair)
            console.log(`Lowlevel address: ${keypair.address}`)
    }
}
f();
```

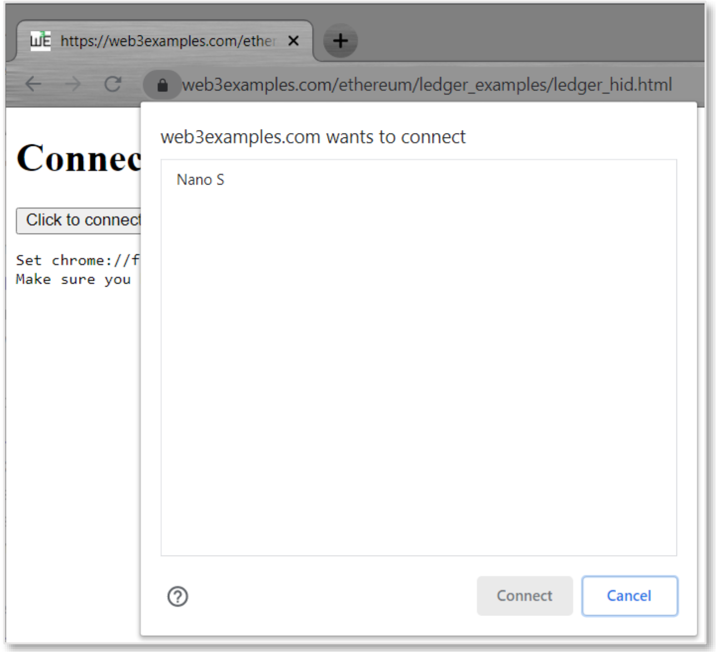
```
> node low_level_ledger.js
Connected to nanoS
Version: 1.6.2
Lowlevel address:0x.....
```

https://github.com/web3examples/ethereum/tree/master/ledger_examples/low_level_ledger.js

PD-12.7 Ledger via WebUSB

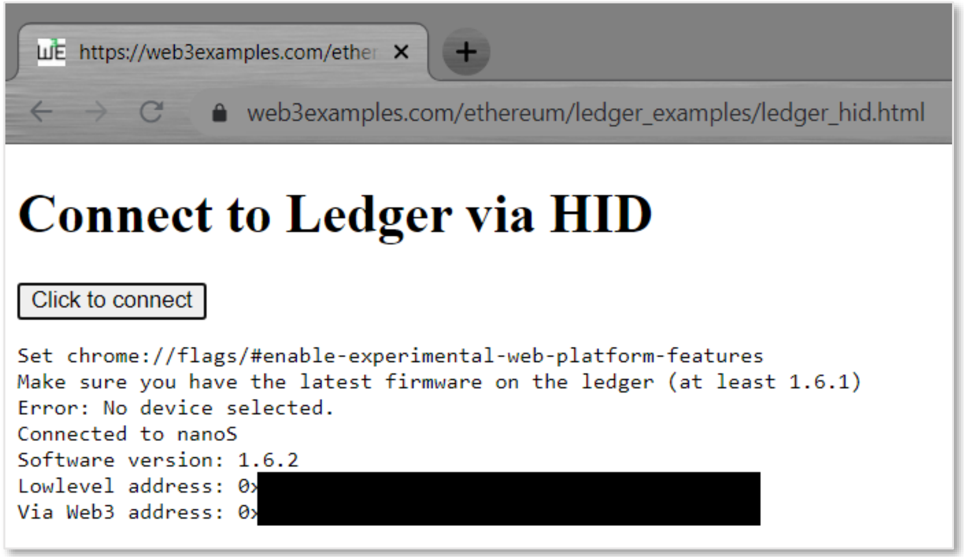


```
<!DOCTYPE html>
<html>
    <head>
        <script src="webusb-browserify.js"></script>  
    </head>
    <body>
        <h1>Connect to Ledger via HID</h1>
        <input type="button" value="Click to connect" onclick="f()">
        <pre id="log" style="width:100%;height:200px"></pre>
<script>
    function log(logstr) {   
            document.getElementById("log").innerHTML +=logstr+"\n";
    }
log("Set chrome://flags/#enable-experimental-web-platform-features");
log("Make sure you have the latest firmware on the ledger (at least 1.6.1)")
async function f() {    
    const transport = await TransportWebUSB.create().catch(x=>log(`Error: ${x.message}`));
    if (transport) {    
        log(`Connected to ${transport.deviceModel.id}`);
        const eth = new AppEth(transport);
        var res=await eth.getAppConfiguration().catch(x=>log(`Error: ${x.message}`));;
        log(`Software version: ${res.version}`);
        var keypair=await eth.getAddress("44'/60'/0'/0/0").catch(x=>log(`Error: ${x.message}`));
        if (keypair)
            log(`Lowlevel address: ${keypair.address}`);
        const engine = new ProviderEngine();
        getTransport=()=> transport;
         const rpcUrl = "https://cloudflare-eth.com"; // "https://ropsten.infura.io/";
        const networkId = 1;
        const ledger = createLedgerSubprovider(getTransport, {
            networkId,
            accountsLength: 1   // nr of accounts retrieved
        });
        engine.addProvider(ledger);
        engine.addProvider(new RpcSubprovider({ rpcUrl }));
        engine.start(); // start polling for blocks
        const web3 = new Web3(engine);
        var acts=await web3.eth.getAccounts().catch(x=>console.log(`Error: ${x.message}`));
        if (acts)        
            log(`Via Web3 address: ${acts[0]}`); // accounts of ledger
    }
}
</script>
```



https://web3examples.com/ether
web3examples.com/ethereum/ledger_examples/ledger_hid.html
Connec
Click to connect
web3examples.com wants to connect
Nano S
Connect
Cancel

```
Set chrome://f
Make sure you
```



https://web3examples.com/ether
web3examples.com/ethereum/ledger_examples/ledger_hid.html
Click to connect

Connect to Ledger via HID

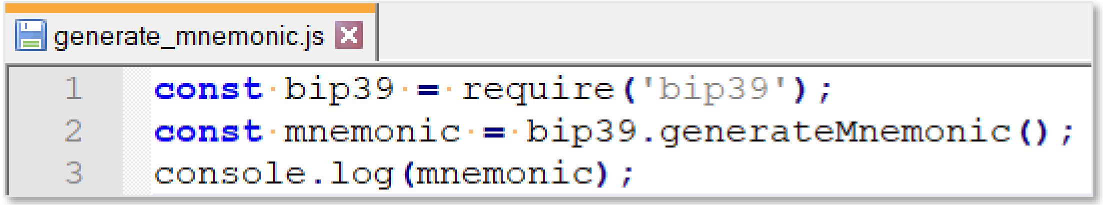
```
Set chrome://flags/#enable-experimental-web-platform-features
Make sure you have the latest firmware on the ledger (at least 1.6.1)
Error: No device selected.
Connected to nanoS
Software version: 1.6.2
Lowlevel address: 0x[redacted]
Via Web3 address: 0x
```

https://web3examples.com/ethereum/ledger_examples/ledger_hid.html

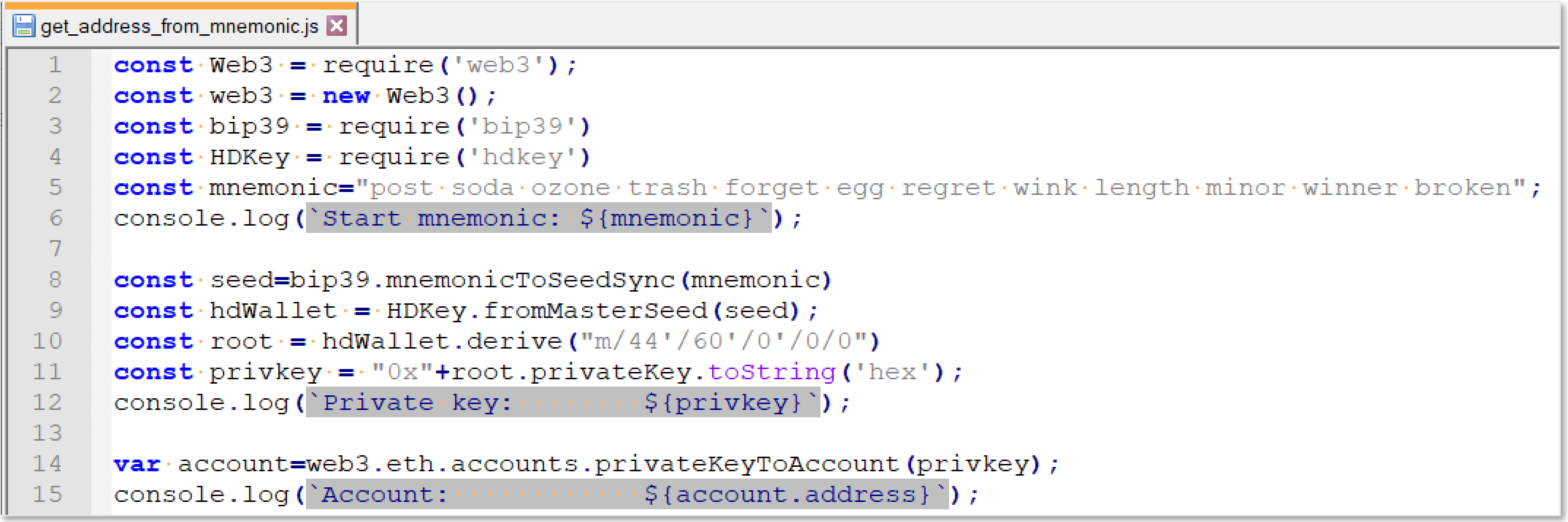
https://github.com/web3examples/ethereum/tree/master/ledger_examples/ledger_hid.html

PD-12.8 Mnemonic & address

```
> npm install bip39
> npm install hdkey
```



```javascript
const bip39 = require('bip39');
const mnemonic = bip39.generateMnemonic();
console.log(mnemonic);
```



```javascript
const Web3 = require('web3');
const web3 = new Web3();
const bip39 = require('bip39')
const HDKey = require('hdkey')
const mnemonic="post soda ozone trash forget egg regret wink length minor winner broken";
console.log(`Start mnemonic: ${mnemonic}`);

const seed=bip39.mnemonicToSeedSync(mnemonic)
const hdWallet = HDKey.fromMasterSeed(seed);
const root = hdWallet.derive("m/44'/60'/0'/0/0")
const privkey = "0x"+root.privateKey.toString('hex');
console.log(`Private key:       ${privkey}`);

var account=web3.eth.accounts.privateKeyToAccount(privkey);
console.log(`Account:           ${account.address}`);
```

```
>node get_address_from_mnemonic.js
Start mnemonic: post soda ozone trash forget egg regret wink length minor winner broken
Private key:      0x04bfcedbbaa686f15643db581857bf06ce19830d10cba4ebf4b35899f1410ad4
Account:          0x6c728716a68499d486cDA1701AB13C7b57f30aA0
```

https://github.com/web3examples/ethereum/tree/master/wallet_examples/generate_mnemonic.js

https://github.com/web3examples/ethereum/tree/master/wallet_examples/get_address_from_mnemonic.js

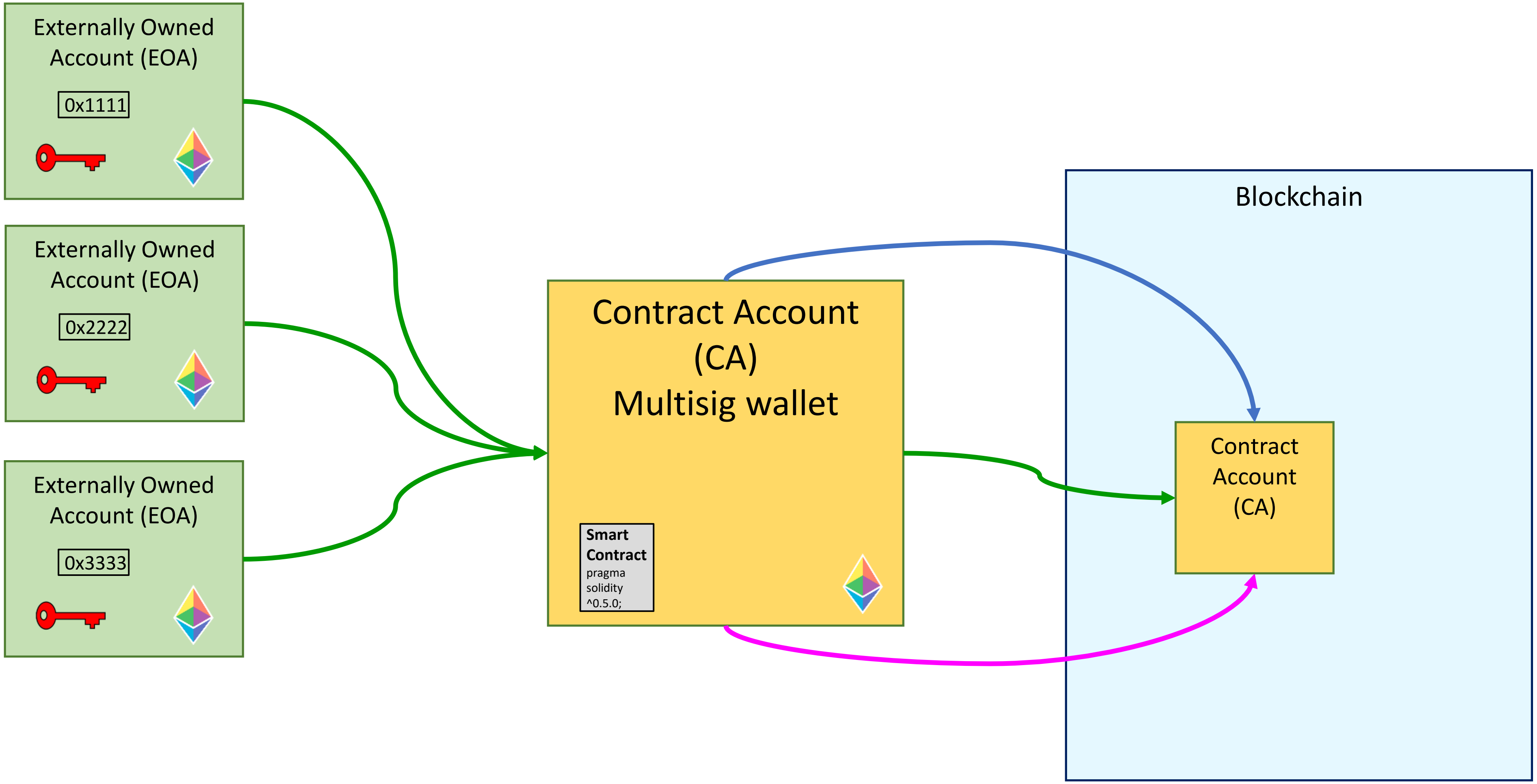
PD-12.9 MultiSigWallet

- Multiple signers required (m of n)
- Time locked transactions / deferred payments
- Limits (per period)
- Freeze / deadman switch / inherit
- Whitelist / Blacklist addresses (policy checking)
- Multiple authentication (2FA)
  - Android
- Pay gas fees in ERC20 tokens
- Batched Transactions

- Recovery methods
  - Recovery via centralized entity
  - Timelocked
  - Social

https://blog.gnosis.pm/smart-wallets-are-here-121d44519cae

PD-12.9 Gnosis Safe Multisig (teams)

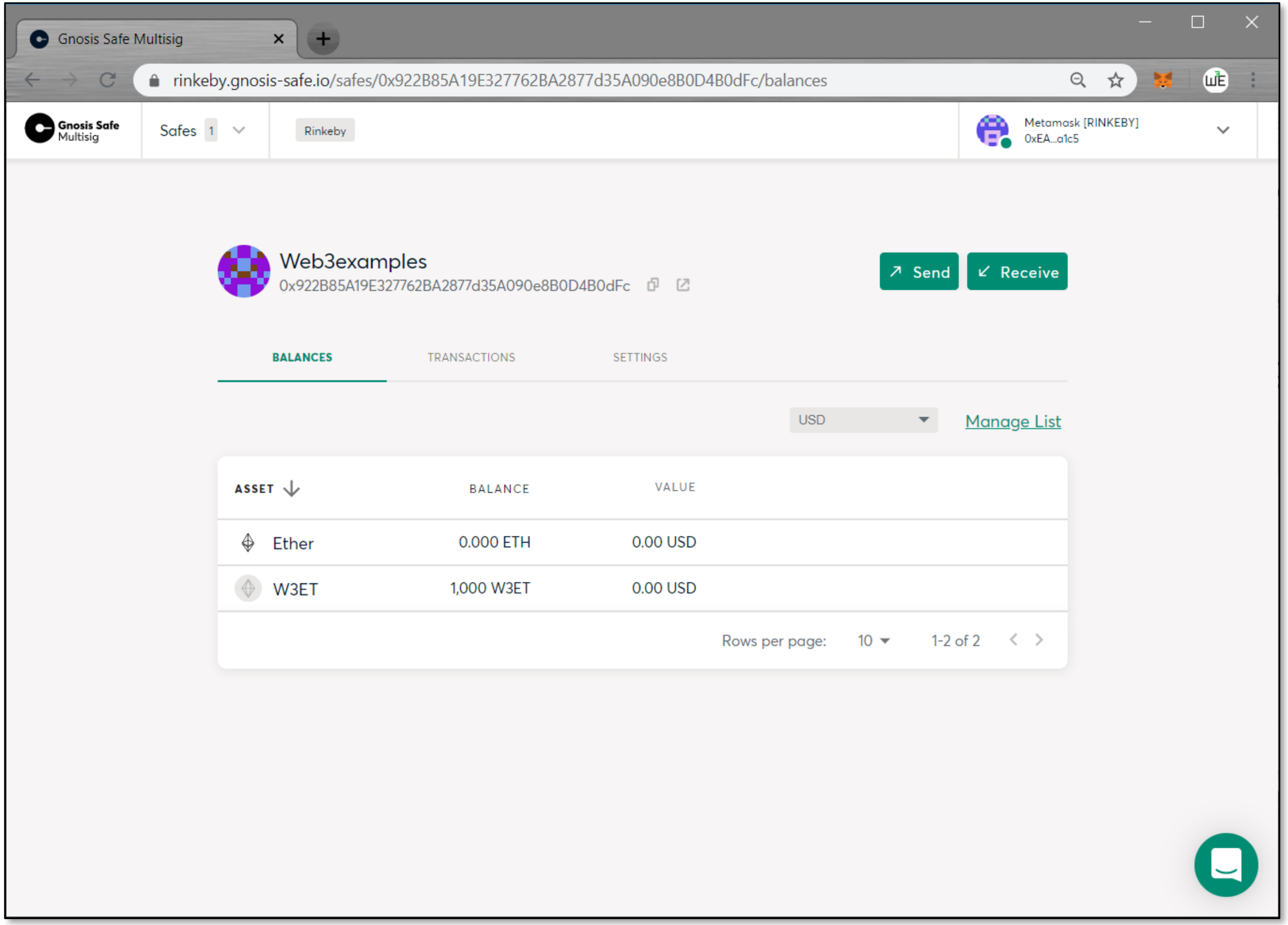


https://gnosis-safe.io

https://rinkeby.gnosis-safe.io

https://docs.gnosis.io/safe

PD-12.9 Gnosis Safe Multisig

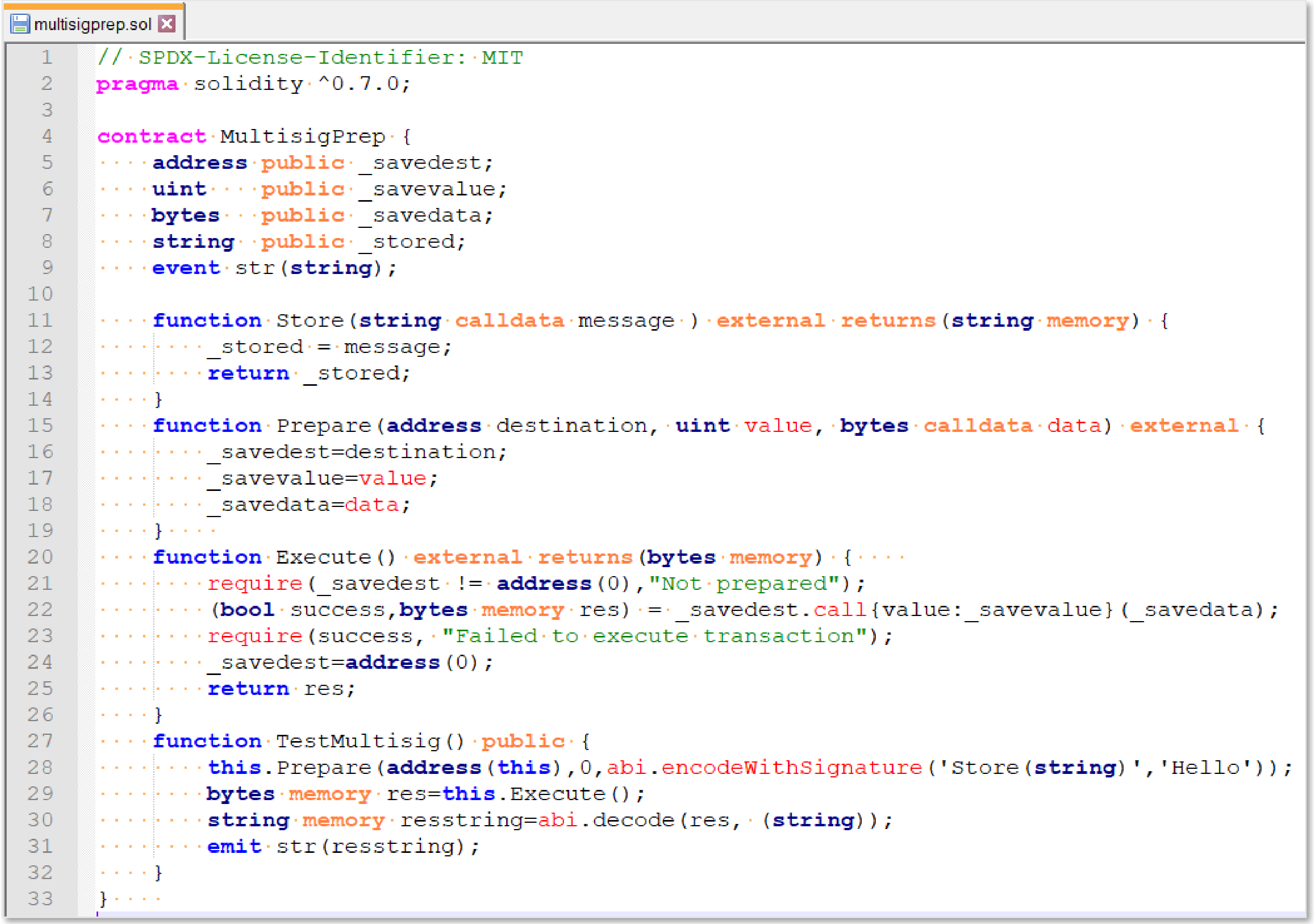


Web3examples
0x922B85A19E327762BA2877d35A090e8B0D4B0dFc

ASSET	BALANCE	VALUE
Ether	0.000 ETH	0.00 USD
W3ET	1,000 W3ET	0.00 USD

https://rinkeby.gnosis-safe.io/safes/0x922B85A19E327762BA2877d35A090e8B0D4B0dFc/balances

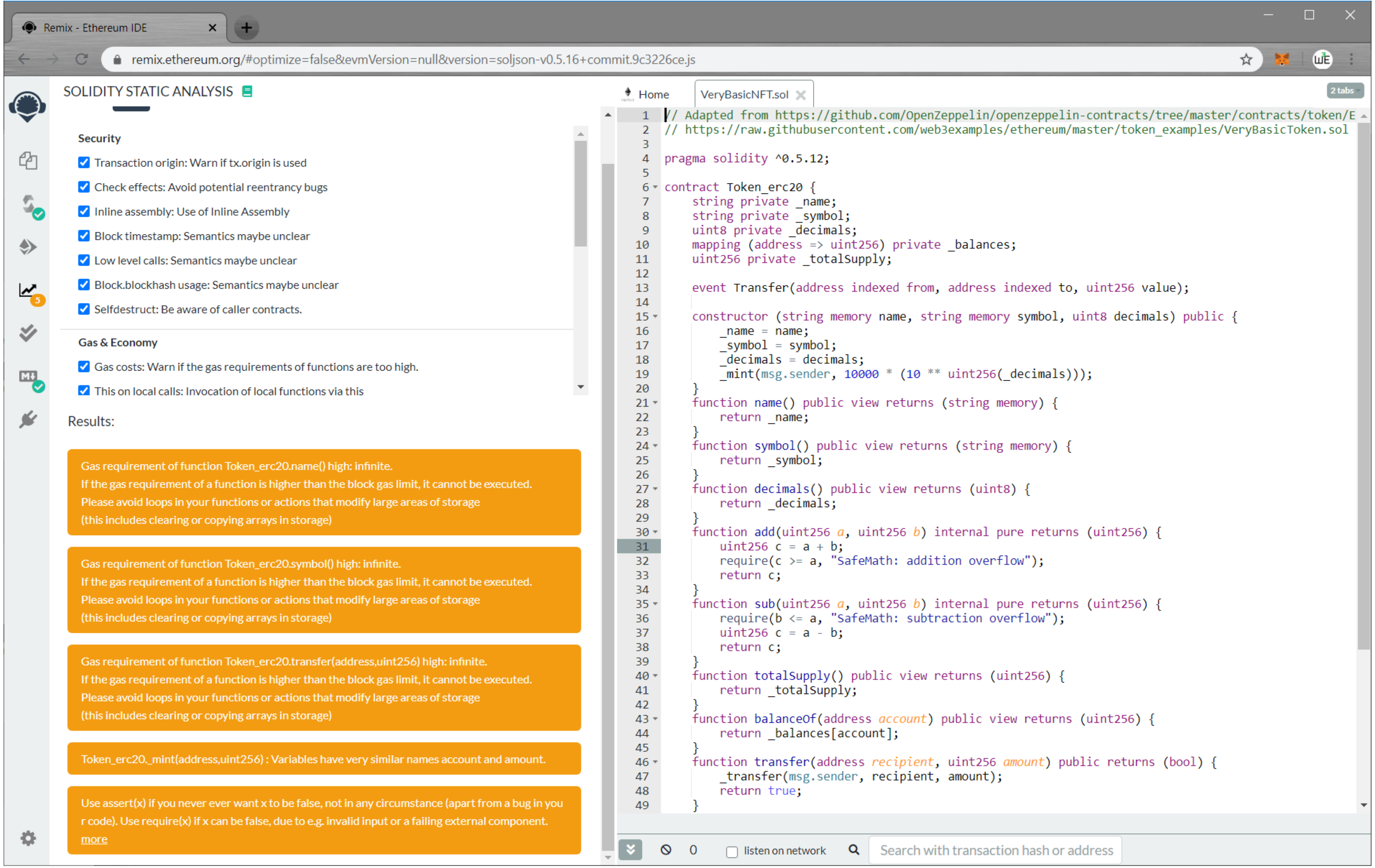
PD-12.9 Multisig Basics



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.7.0;

contract MultisigPrep {
    address public _savedest;
    uint   public _savevalue;
    bytes  public _savedata;
    string public _stored;
    event str(string);

    function Store(string calldata message ) external returns(string memory) {
        _stored = message;
        return _stored;
    }
    function Prepare(address destination, uint value, bytes calldata data) external {
        _savedest=destination;
        _savevalue=value;
        _savedata=data;
    }
    function Execute() external returns(bytes memory) {
        require(_savedest != address(0),"Not prepared");
        (bool success,bytes memory res) = _savedest.call{value:_savevalue}(_savedata);
        require(success, "Failed to execute transaction");
        _savedest=address(0);
        return res;
    }
    function TestMultisig() public {
        this.Prepare(address(this),0,abi.encodeWithSignature('Store(string)','Hello'));
        bytes memory res=this.Execute();
        string memory resstring=abi.decode(res, (string));
        emit str(resstring);
    }
}
```

https://github.com/web3examples/ethereum/tree/master/wallet_examples/multisigprep.sol

PD-12.10 Security tools: Remix: Static analysis



Remix - Ethereum IDE
remix.ethereum.org/#optimize=false&evmVersion=null&version=soljson-v0.5.16+commit.9c3226ce.js
Home
VeryBasicNFT.sol
2 tabs
0
listen on network
Search with transaction hash or address

SOLIDITY STATIC ANALYSIS

Security

- ☑ Transaction origin: Warn if tx.origin is used
- ☑ Check effects: Avoid potential reentrancy bugs
- ☑ Inline assembly: Use of Inline Assembly
- ☑ Block timestamp: Semantics maybe unclear
- ☑ Low level calls: Semantics maybe unclear
- ☑ Block.blockhash usage: Semantics maybe unclear
- ☑ Selfdestruct: Be aware of caller contracts.

Gas & Economy

- ☑ Gas costs: Warn if the gas requirements of functions are too high.
- ☑ This on local calls: Invocation of local functions via this

Results:

Gas requirement of function Token_erc20.name() high: infinite.
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)

Gas requirement of function Token_erc20.symbol() high: infinite.
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)

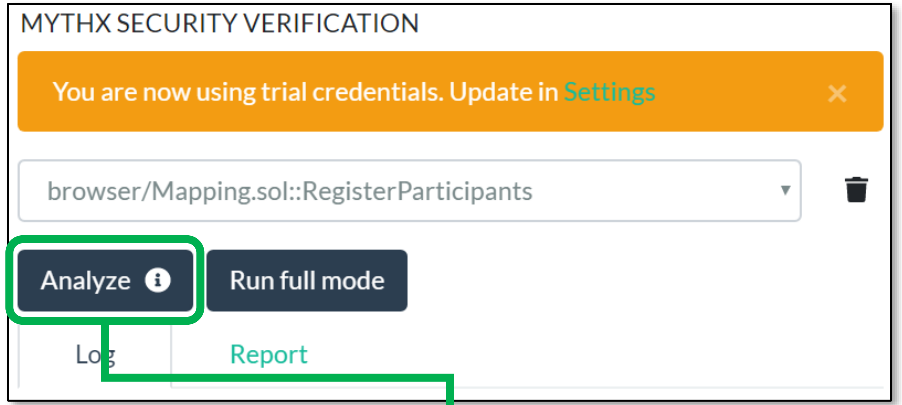
Gas requirement of function Token_erc20.transfer(address,uint256) high: infinite.
If the gas requirement of a function is higher than the block gas limit, it cannot be executed.
Please avoid loops in your functions or actions that modify large areas of storage
(this includes clearing or copying arrays in storage)

Token_erc20._mint(address,uint256) : Variables have very similar names account and amount.

Use assert(x) if you never ever want x to be false, not in any circumstance (apart from a bug in your code). Use require(x) if x can be false, due to e.g. invalid input or a failing external component.
more

```
// Adapted from https://github.com/OpenZeppelin/openzeppelin-contracts/tree/master/contracts/token/E
// https://raw.githubusercontent.com/web3examples/ethereum/master/token_examples/VeryBasicToken.sol

pragma solidity ^0.5.12;

contract Token_erc20 {
    string private _name;
    string private _symbol;
    uint8 private _decimals;
    mapping (address => uint256) private _balances;
    uint256 private _totalSupply;

    event Transfer(address indexed from, address indexed to, uint256 value);

    constructor (string memory name, string memory symbol, uint8 decimals) public {
        _name = name;
        _symbol = symbol;
        _decimals = decimals;
        _mint(msg.sender, 10000 * (10 ** uint256(_decimals)));
    }
    function name() public view returns (string memory) {
        return _name;
    }
    function symbol() public view returns (string memory) {
        return _symbol;
    }
    function decimals() public view returns (uint8) {
        return _decimals;
    }
    function add(uint256 a, uint256 b) internal pure returns (uint256) {
        uint256 c = a + b;
        require(c >= a, "SafeMath: addition overflow");
        return c;
    }
    function sub(uint256 a, uint256 b) internal pure returns (uint256) {
        require(b <= a, "SafeMath: subtraction overflow");
        uint256 c = a - b;
        return c;
    }
    function totalSupply() public view returns (uint256) {
        return _totalSupply;
    }
    function balanceOf(address account) public view returns (uint256) {
        return _balances[account];
    }
    function transfer(address recipient, uint256 amount) public returns (bool) {
        _transfer(msg.sender, recipient, amount);
        return true;
    }
```

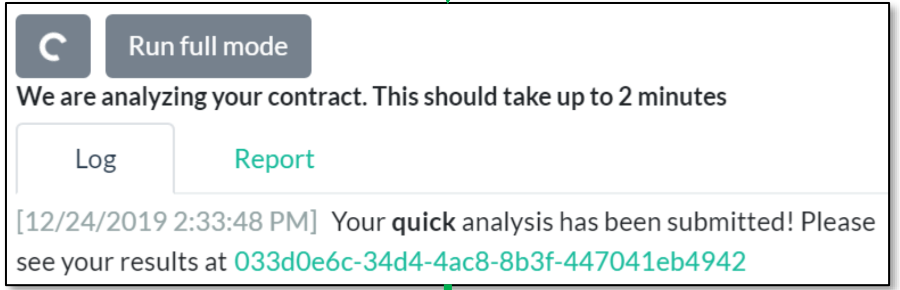
PD-12.10 Security tests Remix: Mythx



MYTHX SECURITY VERIFICATION

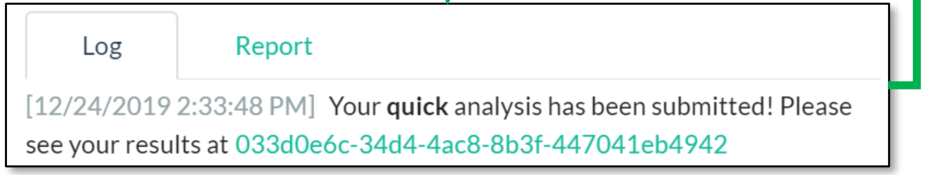
You are now using trial credentials. Update in Settings ×

browser/Mapping.sol::RegisterParticipants ▾

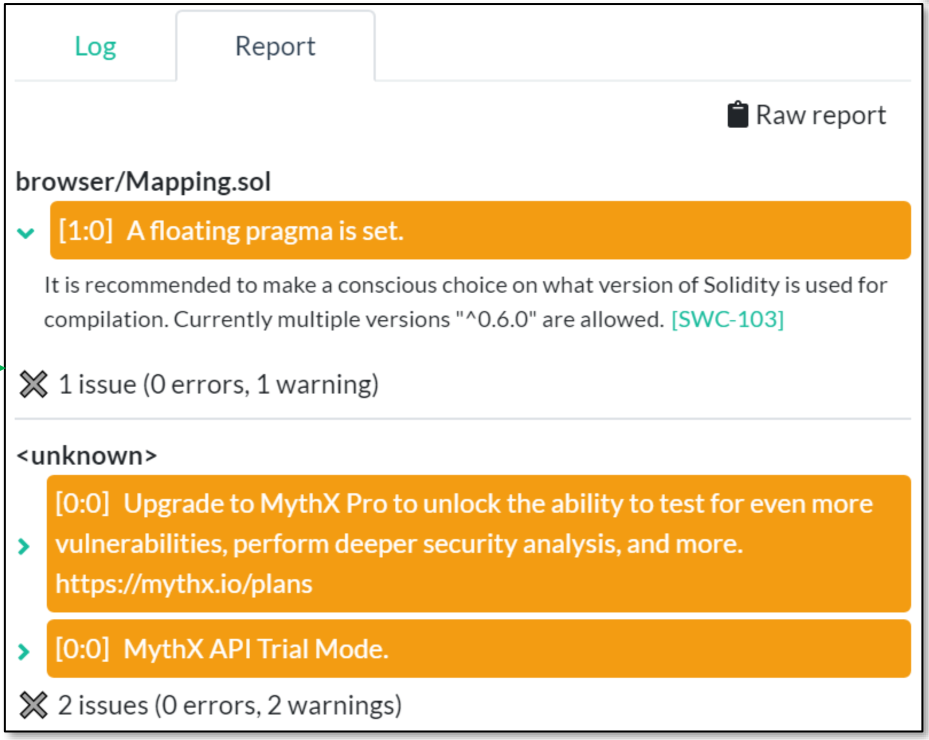


We are analyzing your contract. This should take up to 2 minutes

[12/24/2019 2:33:48 PM] Your **quick** analysis has been submitted! Please see your results at 033d0e6c-34d4-4ac8-8b3f-447041eb4942



[12/24/2019 2:33:48 PM] Your **quick** analysis has been submitted! Please see your results at 033d0e6c-34d4-4ac8-8b3f-447041eb4942



browser/Mapping.sol

[1:0] A floating pragma is set.

It is recommended to make a conscious choice on what version of Solidity is used for compilation. Currently multiple versions "^0.6.0" are allowed. [SWC-103]

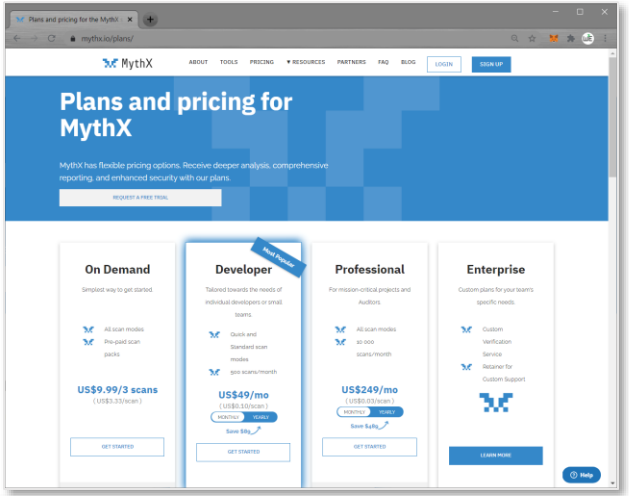
1 issue (0 errors, 1 warning)

<unknown>

[0:0] Upgrade to MythX Pro to unlock the ability to test for even more vulnerabilities, perform deeper security analysis, and more. https://mythx.io/plans

[0:0] MythX API Trial Mode.

2 issues (0 errors, 2 warnings)



Plans and pricing for MythX

MythX has flexible pricing options. Receive deeper analysis, comprehensive reporting, and enhanced security with our plans.

On Demand — Simplest way to get started. All scan modes; Pre-paid scan packs. US$9.99/3 scans (US$3.33/scan) GET STARTED

Developer (Most Popular) — Tailored towards the needs of individual developers or small teams. Quick and Standard scan modes; 500 scans/month. US$49/mo (US$0.10/scan) MONTHLY YEARLY Save $89 GET STARTED

Professional — For mission-critical projects and Auditors. All scan modes; 10 000 scans/month. US$249/mo (US$0.03/scan) MONTHLY YEARLY Save $489 GET STARTED

Enterprise — Custom plans for your team's specific needs. Custom Verification Service; Retainer for Custom Support. LEARN MORE

https://mythx.io/

https://dashboard.mythx.io/#/login

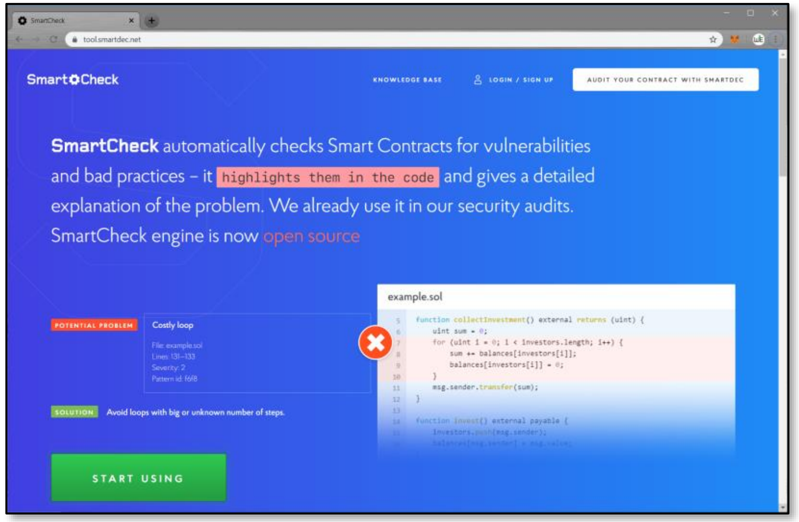
https://docs.mythx.io/en/latest/tools/

https://docs.mythx.io/en/latest/tools/remix/

https://github.com/aquiladev/remix-mythx-plugin

https://blog.mythx.io/howto/a-beginners-guide-to-mythx/

PD-12.10 Security tests SmartCheck



SmartCheck automatically checks Smart Contracts for vulnerabilities and bad practices – it `highlights them in the code` and gives a detailed explanation of the problem. We already use it in our security audits. SmartCheck engine is now open source

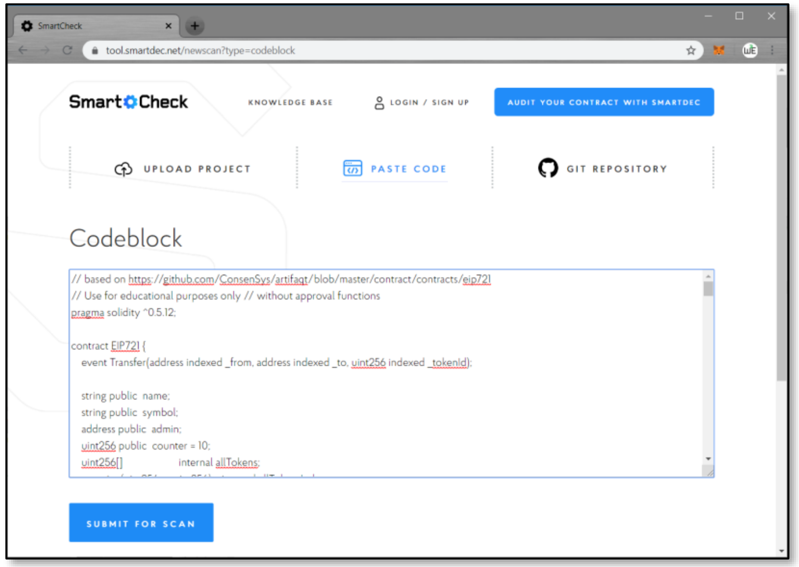
POTENTIAL PROBLEM Costly loop

File: example.sol
Lines: 131–133
Severity: 2
Pattern id: f6f8

SOLUTION Avoid loops with big or unknown number of steps.

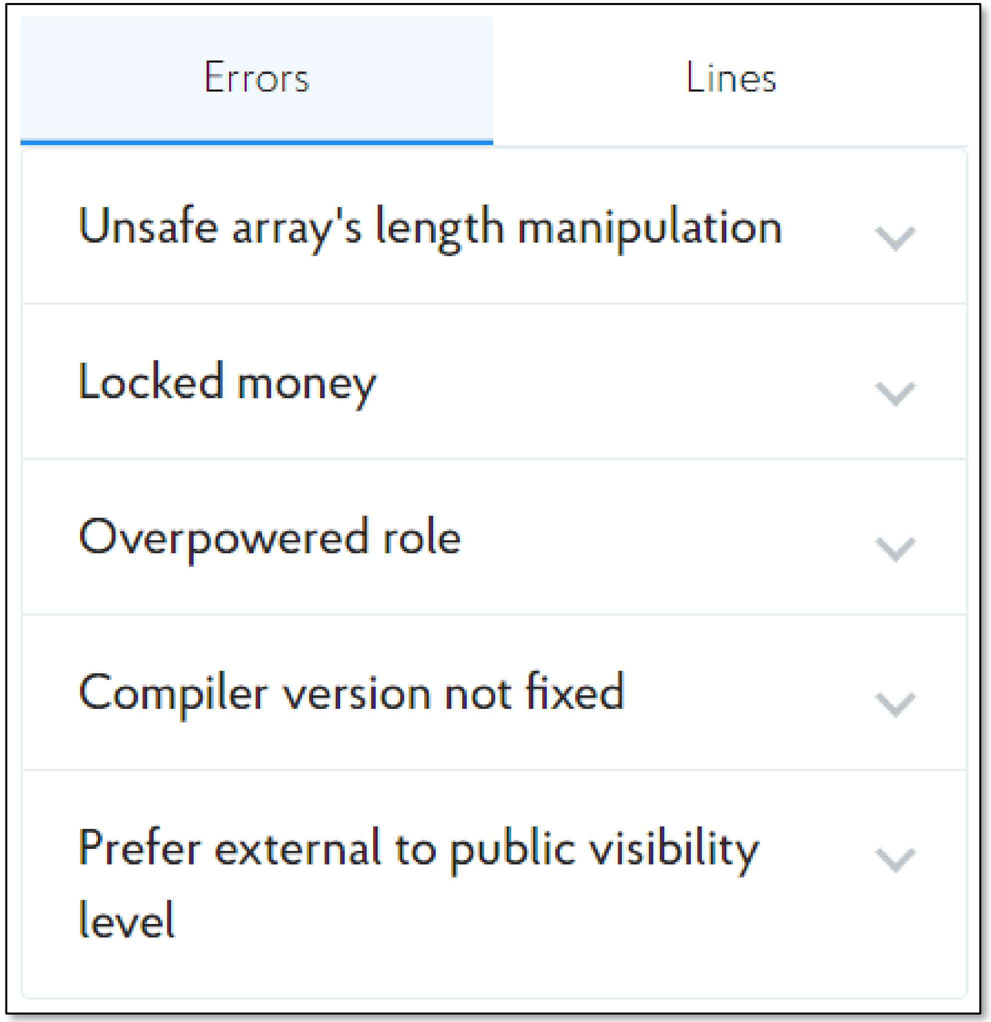
example.sol

```
function collectInvestment() external returns (uint) {
    uint sum = 0;
    for (uint i = 0; i < investors.length; i++) {
        sum += balances[investors[i]];
        balances[investors[i]] = 0;
    }
    msg.sender.transfer(sum);
}

function invest() external payable {
    investors.push(msg.sender);
    balances[msg.sender] = msg.value;
```



Codeblock

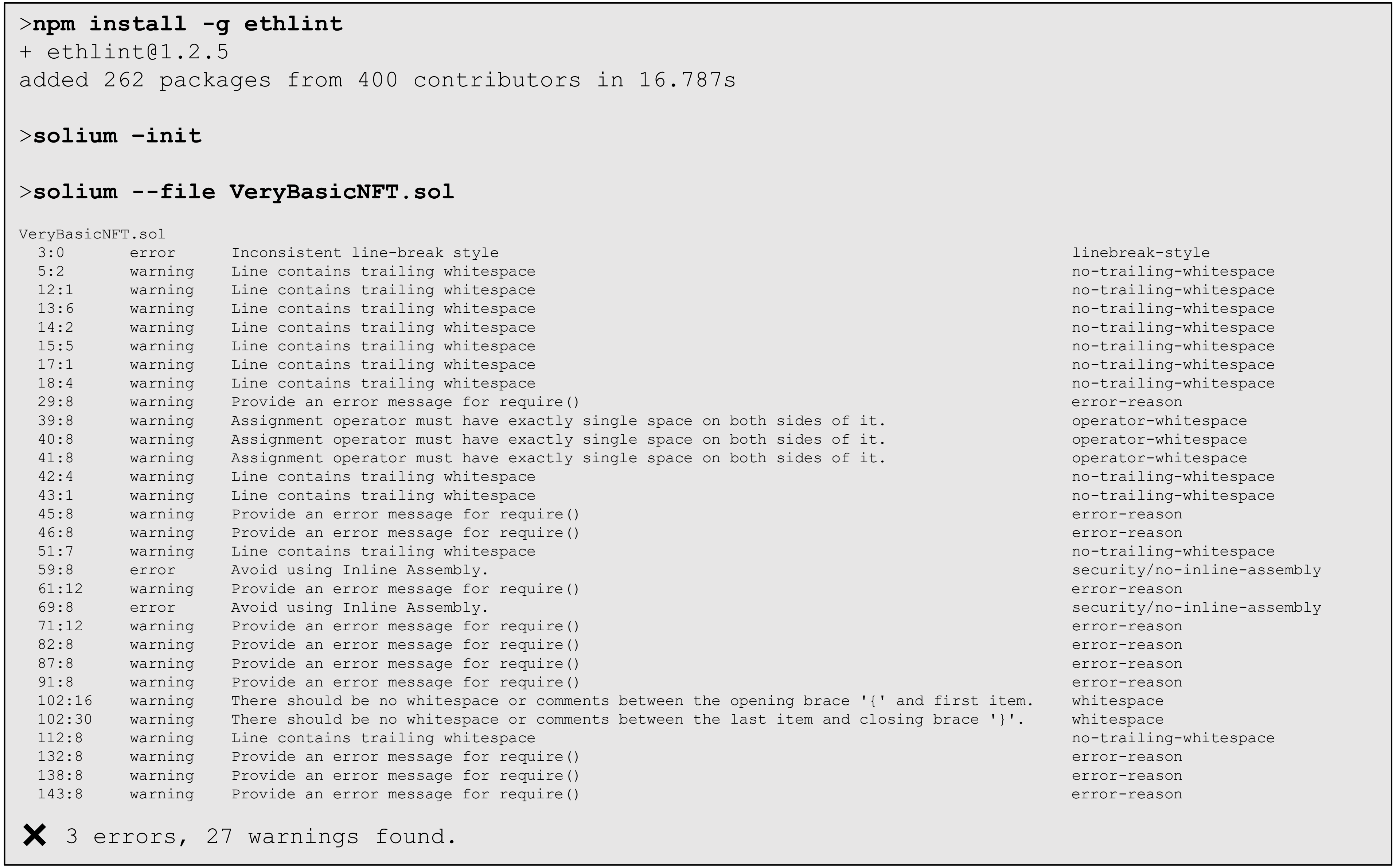
```
// based on https://github.com/ConsenSys/artifaqt/blob/master/contract/contracts/eip721
// Use for educational purposes only // without approval functions
pragma solidity ^0.5.12;

contract EIP721 {
  event Transfer(address indexed _from, address indexed _to, uint256 indexed _tokenId);

  string public  name;
  string public  symbol;
  address public  admin;
  uint256 public  counter = 10;
  uint256[]              internal allTokens;
```



- Unsafe array's length manipulation
- Locked money
- Overpowered role
- Compiler version not fixed
- Prefer external to public visibility level

https://tool.smartdec.net/

PD-12.10 Security tests: ethlint

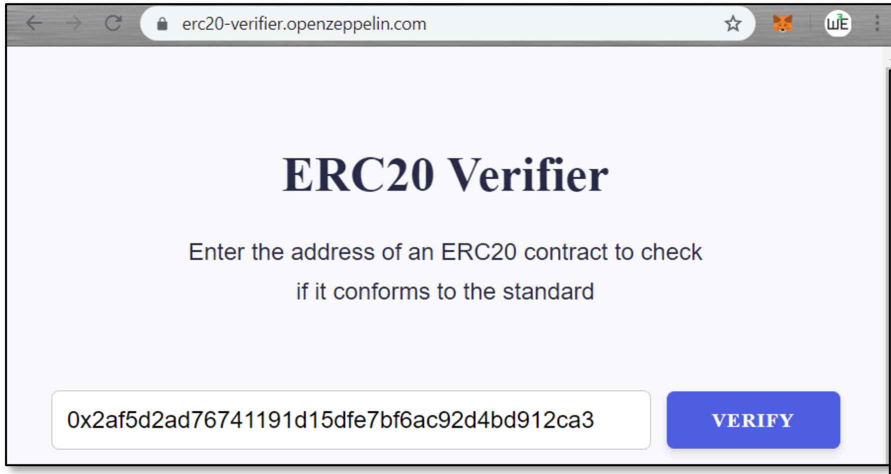


```
>npm install -g ethlint
+ ethlint@1.2.5
added 262 packages from 400 contributors in 16.787s

>solium -init

>solium --file VeryBasicNFT.sol

VeryBasicNFT.sol
  3:0       error      Inconsistent line-break style                                                          linebreak-style
  5:2       warning    Line contains trailing whitespace                                                      no-trailing-whitespace
  12:1      warning    Line contains trailing whitespace                                                      no-trailing-whitespace
  13:6      warning    Line contains trailing whitespace                                                      no-trailing-whitespace
  14:2      warning    Line contains trailing whitespace                                                      no-trailing-whitespace
  15:5      warning    Line contains trailing whitespace                                                      no-trailing-whitespace
  17:1      warning    Line contains trailing whitespace                                                      no-trailing-whitespace
  18:4      warning    Line contains trailing whitespace                                                      no-trailing-whitespace
  29:8      warning    Provide an error message for require()                                                 error-reason
  39:8      warning    Assignment operator must have exactly single space on both sides of it.               operator-whitespace
  40:8      warning    Assignment operator must have exactly single space on both sides of it.               operator-whitespace
  41:8      warning    Assignment operator must have exactly single space on both sides of it.               operator-whitespace
  42:4      warning    Line contains trailing whitespace                                                      no-trailing-whitespace
  43:1      warning    Line contains trailing whitespace                                                      no-trailing-whitespace
  45:8      warning    Provide an error message for require()                                                 error-reason
  46:8      warning    Provide an error message for require()                                                 error-reason
  51:7      warning    Line contains trailing whitespace                                                      no-trailing-whitespace
  59:8      error      Avoid using Inline Assembly.                                                           security/no-inline-assembly
  61:12     warning    Provide an error message for require()                                                 error-reason
  69:8      error      Avoid using Inline Assembly.                                                           security/no-inline-assembly
  71:12     warning    Provide an error message for require()                                                 error-reason
  82:8      warning    Provide an error message for require()                                                 error-reason
  87:8      warning    Provide an error message for require()                                                 error-reason
  91:8      warning    Provide an error message for require()                                                 error-reason
  102:16    warning    There should be no whitespace or comments between the opening brace '{' and first item.    whitespace
  102:30    warning    There should be no whitespace or comments between the last item and closing brace '}'.     whitespace
  112:8     warning    Line contains trailing whitespace                                                      no-trailing-whitespace
  132:8     warning    Provide an error message for require()                                                 error-reason
  138:8     warning    Provide an error message for require()                                                 error-reason
  143:8     warning    Provide an error message for require()                                                 error-reason

✖ 3 errors, 27 warnings found.
```

https://ethlint.readthedocs.io

https://github.com/duaraghav8/Ethlint

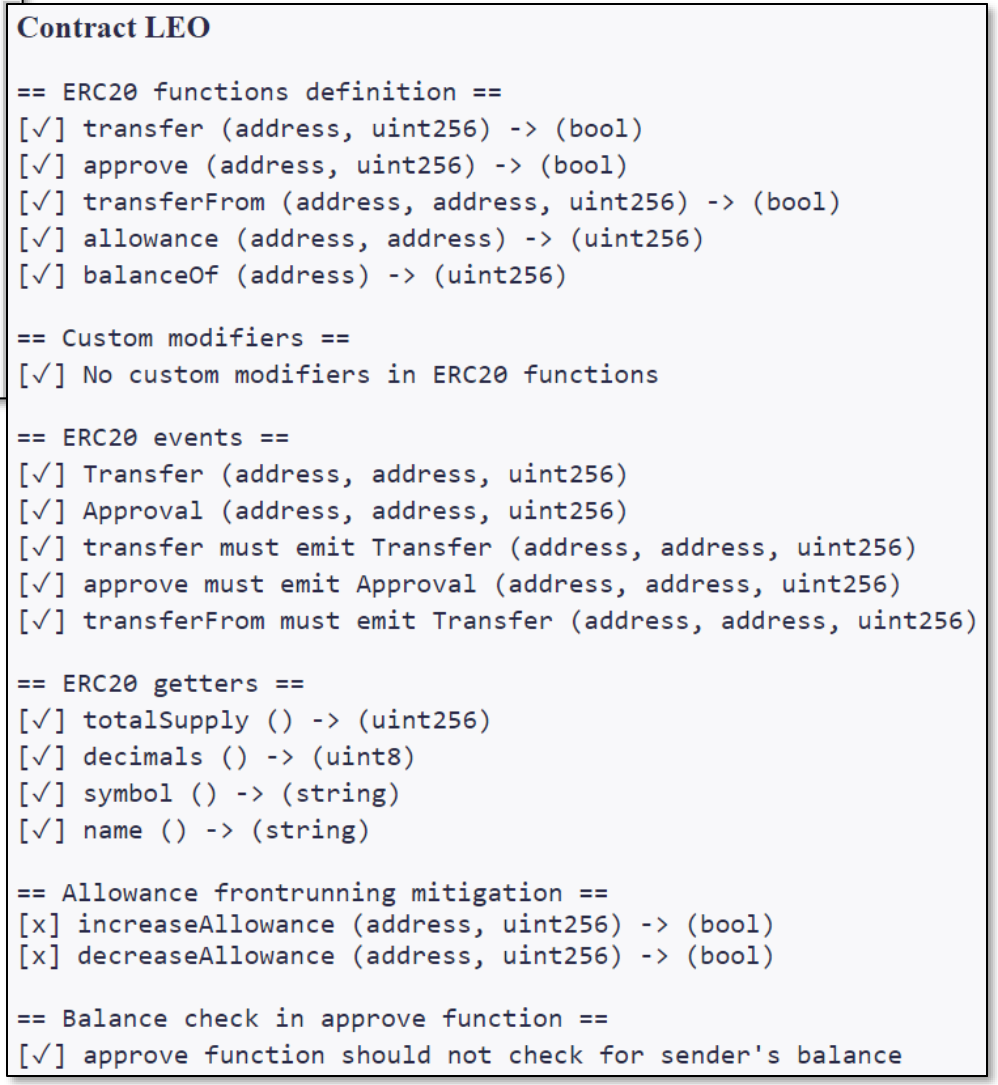
PD-12.10 Security tests: ERC20 verifier



ERC20 Verifier

Enter the address of an ERC20 contract to check
if it conforms to the standard

0x2af5d2ad76741191d15dfe7bf6ac92d4bd912ca3

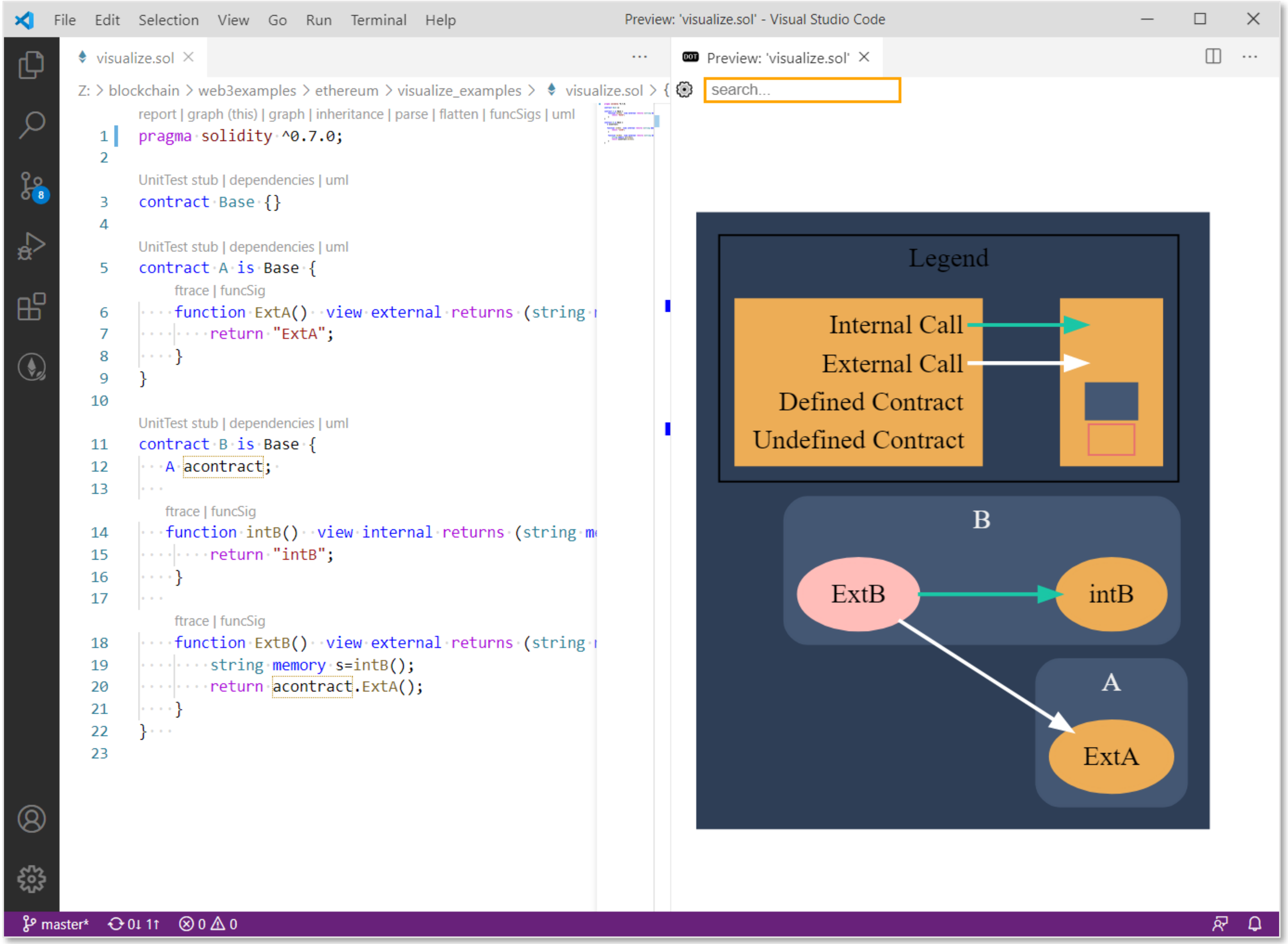


Contract LEO

```
== ERC20 functions definition ==
[√] transfer (address, uint256) -> (bool)
[√] approve (address, uint256) -> (bool)
[√] transferFrom (address, address, uint256) -> (bool)
[√] allowance (address, address) -> (uint256)
[√] balanceOf (address) -> (uint256)

== Custom modifiers ==
[√] No custom modifiers in ERC20 functions

== ERC20 events ==
[√] Transfer (address, address, uint256)
[√] Approval (address, address, uint256)
[√] transfer must emit Transfer (address, address, uint256)
[√] approve must emit Approval (address, address, uint256)
[√] transferFrom must emit Transfer (address, address, uint256)

== ERC20 getters ==
[√] totalSupply () -> (uint256)
[√] decimals () -> (uint8)
[√] symbol () -> (string)
[√] name () -> (string)

== Allowance frontrunning mitigation ==
[x] increaseAllowance (address, uint256) -> (bool)
[x] decreaseAllowance (address, uint256) -> (bool)

== Balance check in approve function ==
[√] approve function should not check for sender's balance
```

https://erc20-verifier.openzeppelin.com/

PD-12.10 VSCode Visualize


File
Edit
Selection
View
Go
Run
Terminal
Help
Preview: 'visualize.sol' - Visual Studio Code
visualize.sol
Preview: 'visualize.sol'
Z: > blockchain > web3examples > ethereum > visualize_examples > visualize.sol > {
search...
report | graph (this) | graph | inheritance | parse | flatten | funcSigs | uml
UnitTest stub | dependencies | uml
ftrace | funcSig
Legend
Internal Call
External Call
Defined Contract
Undefined Contract
B
ExtB
intB
A
ExtA
master*
0↓ 1↑
⊗ 0 ⚠ 0

```
pragma solidity ^0.7.0;

contract Base {}

contract A is Base {
    function ExtA()  view external returns (string 
        return "ExtA";
    }
}

contract B is Base {
  A acontract;

  function intB()  view internal returns (string m
        return "intB";
    }

    function ExtB()  view external returns (string 
        string memory s=intB();
        return acontract.ExtA();
    }
}
```

https://code.visualstudio.com

https://marketplace.visualstudio.com/items?itemName=tintinweb.solidity-visual-auditor

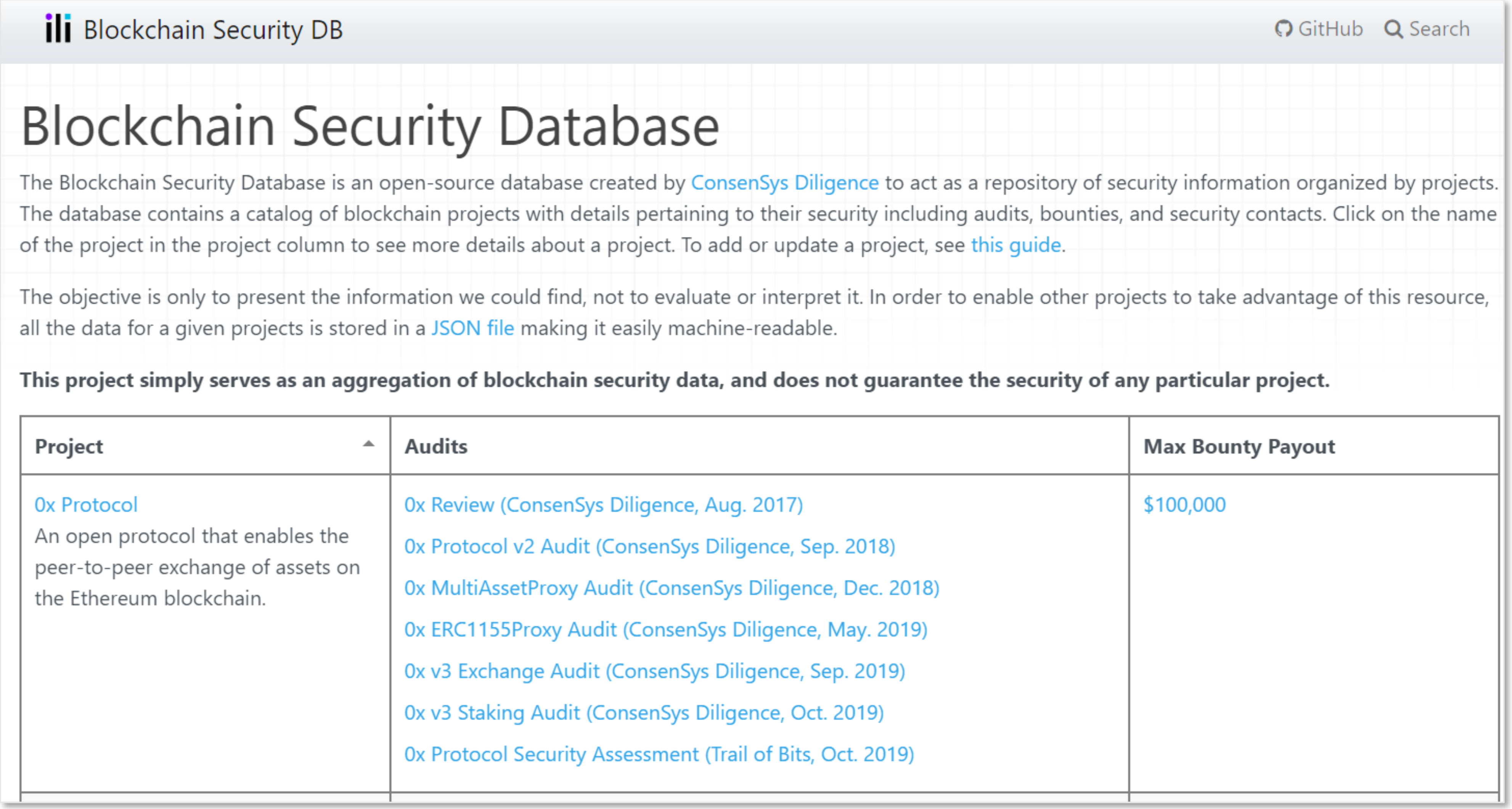
https://github.com/web3examples/ethereum/blob/master/visualize_examples/visualize.sol

PD-12.11 Audits, Bounties & challenges

Serial Number	Audit Class	Audit Subclass
1	Overflow Audit	-
2	Race Conditions Audit	-
3	Permission Vulnerability Audit	Authority Vulnerability Audit
		Excessive auditing authority Audit
4	Safety Design Audit	Zeppelin Module Safe Use Audit
		Compiler Version Security Audit
		Hard-coded Address Security Audit
		Fallback Function Safe Use Audit
		Show Coding Security Audit
		Function Return Value Security Audit
		Call Function Security Audit
5	Denial of Service Audit	-
6	Gas Optimization Audit	-
7	Design Logic Audit	-
8	"False-Deposit" Vulnerability Audit	-
9	Malicious Event Log Audit	-
10	Scoping and Declarations Audit	-
11	Replay Attack Audit	ECDSA's Signature Replay Audit
12	Uninitialized Storage Pointer Audit	-
13	Arithmetic Accuracy Deviation Audit	-

https://www.slowmist.com/en/service-smart-contract-security-audit.html

PD-12.11 Audits: Blockchain Security Database



Blockchain Security Database

The Blockchain Security Database is an open-source database created by ConsenSys Diligence to act as a repository of security information organized by projects. The database contains a catalog of blockchain projects with details pertaining to their security including audits, bounties, and security contacts. Click on the name of the project in the project column to see more details about a project. To add or update a project, see this guide.

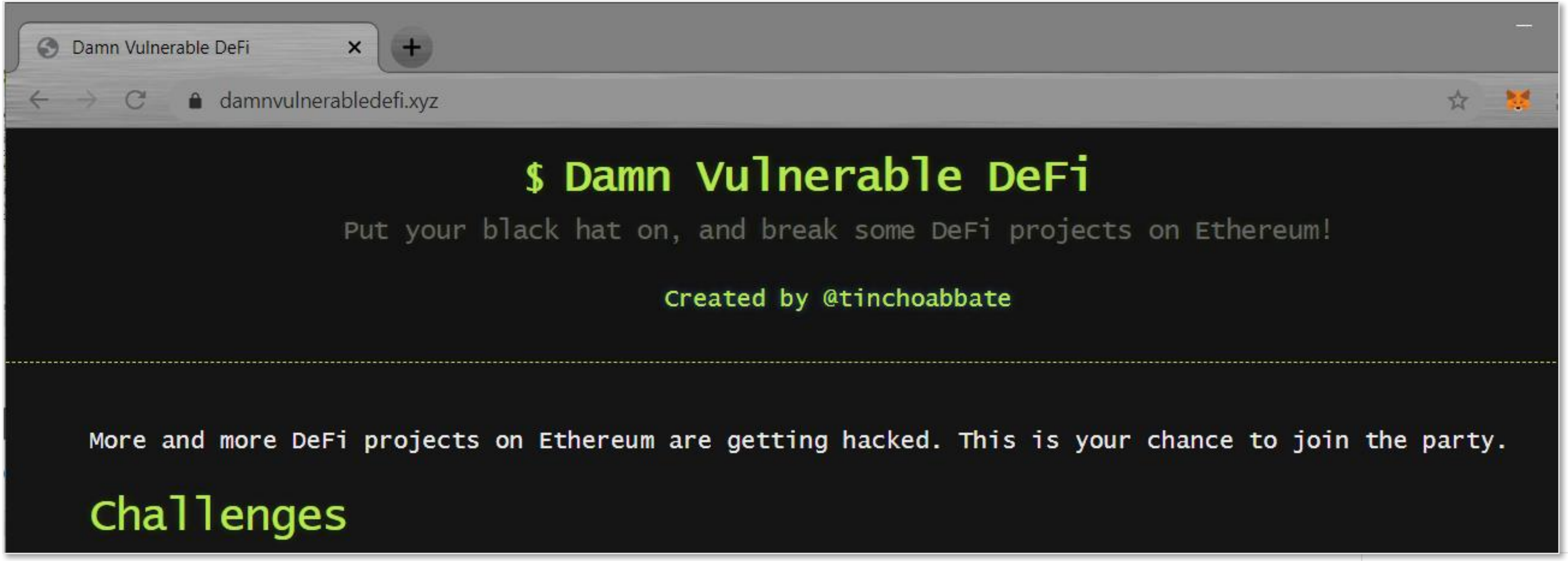
The objective is only to present the information we could find, not to evaluate or interpret it. In order to enable other projects to take advantage of this resource, all the data for a given projects is stored in a JSON file making it easily machine-readable.

This project simply serves as an aggregation of blockchain security data, and does not guarantee the security of any particular project.

Project	Audits	Max Bounty Payout
0x Protocol An open protocol that enables the peer-to-peer exchange of assets on the Ethereum blockchain.	0x Review (ConsenSys Diligence, Aug. 2017) 0x Protocol v2 Audit (ConsenSys Diligence, Sep. 2018) 0x MultiAssetProxy Audit (ConsenSys Diligence, Dec. 2018) 0x ERC1155Proxy Audit (ConsenSys Diligence, May. 2019) 0x v3 Exchange Audit (ConsenSys Diligence, Sep. 2019) 0x v3 Staking Audit (ConsenSys Diligence, Oct. 2019) 0x Protocol Security Assessment (Trail of Bits, Oct. 2019)	$100,000

https://consensys.github.io/blockchainSecurityDB

PD-12.11 Bounties & challenges



$ Damn Vulnerable DeFi

Put your black hat on, and break some DeFi projects on Ethereum!

Created by @tinchoabbate

More and more DeFi projects on Ethereum are getting hacked. This is your chance to join the party.

Challenges

https://www.damnvulnerabledefi.xyz

https://explorer.bounties.network/explorer

https://beta.bounty0x.io/explore

https://ethereum.org/en/eth2/get-involved/bug-bounty

https://bounty.ethereum.org

https://immunefi.com