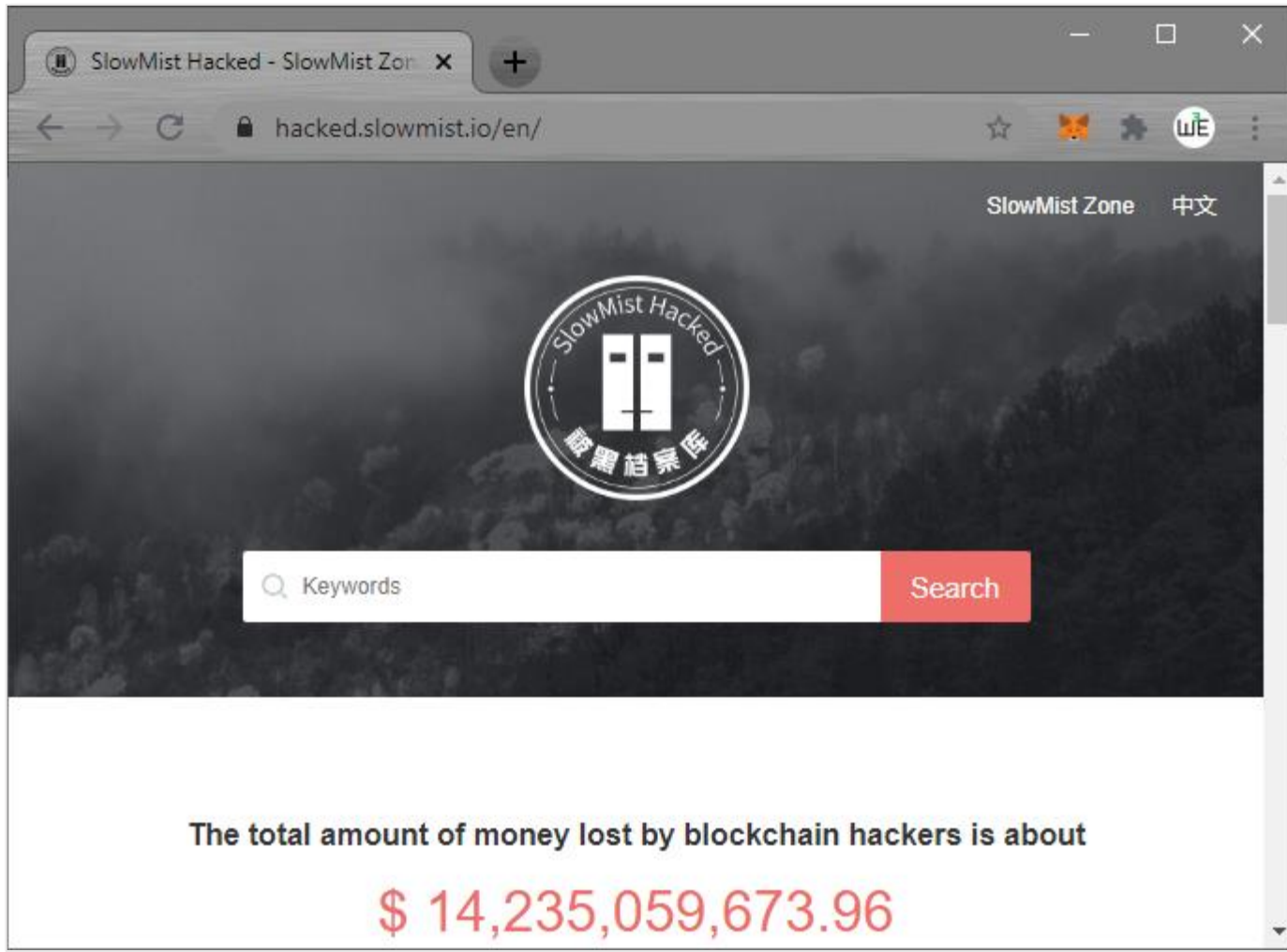


PD-12.0 Security

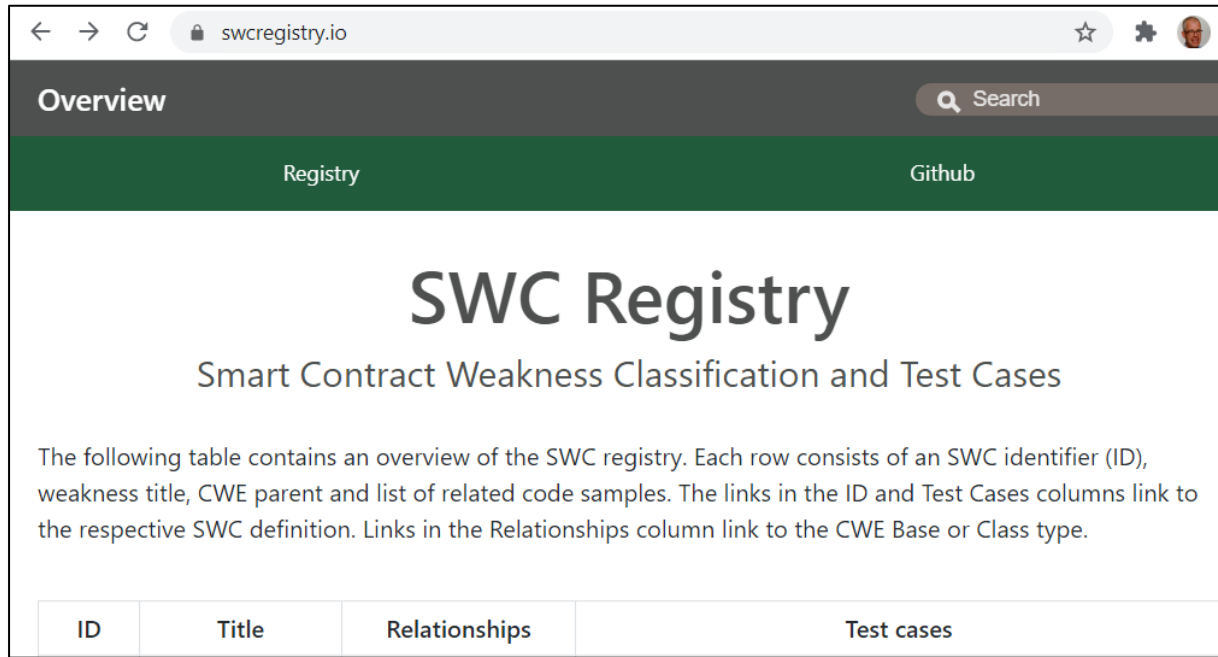


PD-12.1 Hacks and Weaknesses



PD-12.1 Smart Contract Weakness Classification Registry

SWC Registry



The screenshot shows a web browser window with the URL swcregistry.io. The page has a dark green header with the word "Overview" on the left and a search bar on the right. Below the header, there are two buttons: "Registry" and "Github". The main content area features the title "SWC Registry" in a large font, followed by the subtitle "Smart Contract Weakness Classification and Test Cases". A paragraph of text explains that the following table contains an overview of the SWC registry, with each row consisting of an SWC identifier (ID), weakness title, CWE parent, and list of related code samples. The text also notes that links in the ID and Test Cases columns link to the respective SWC definition, and links in the Relationships column link to the CWE Base or Class type. Below this text is a table with four columns: "ID", "Title", "Relationships", and "Test cases".

ID	Title	Relationships	Test cases
----	-------	---------------	------------



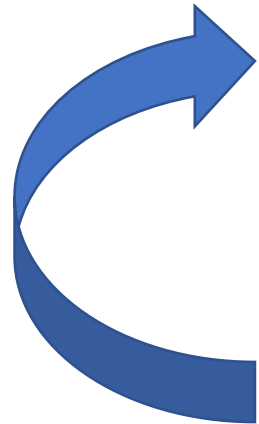
<https://cwe.mitre.org/>

<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();  
}
```

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

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

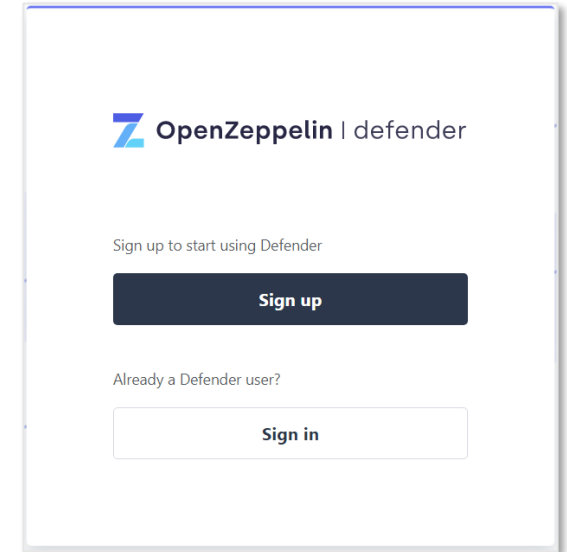
Security Tools

Bug Bounty Programs

About ▾

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



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

PD-12.4 Access control 1/2: Ownable

```
whitelist.sol x
1 // based on: https://docs.openzeppelin.com/contracts/3.x/access-control
2 // SPDX-License-Identifier: MIT
3 pragma solidity ^0.6.0;
4
5 import "@openzeppelin/contracts/access/Ownable.sol";
6 contract Whitelist is Ownable {
7     mapping(address => bool) members;
8
9     constructor() public Ownable() {
10     }
11
12     function addMember(address _member)
13     public
14     onlyOwner
15     {
16         members[_member] = true;
17     }
18 }
```

<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

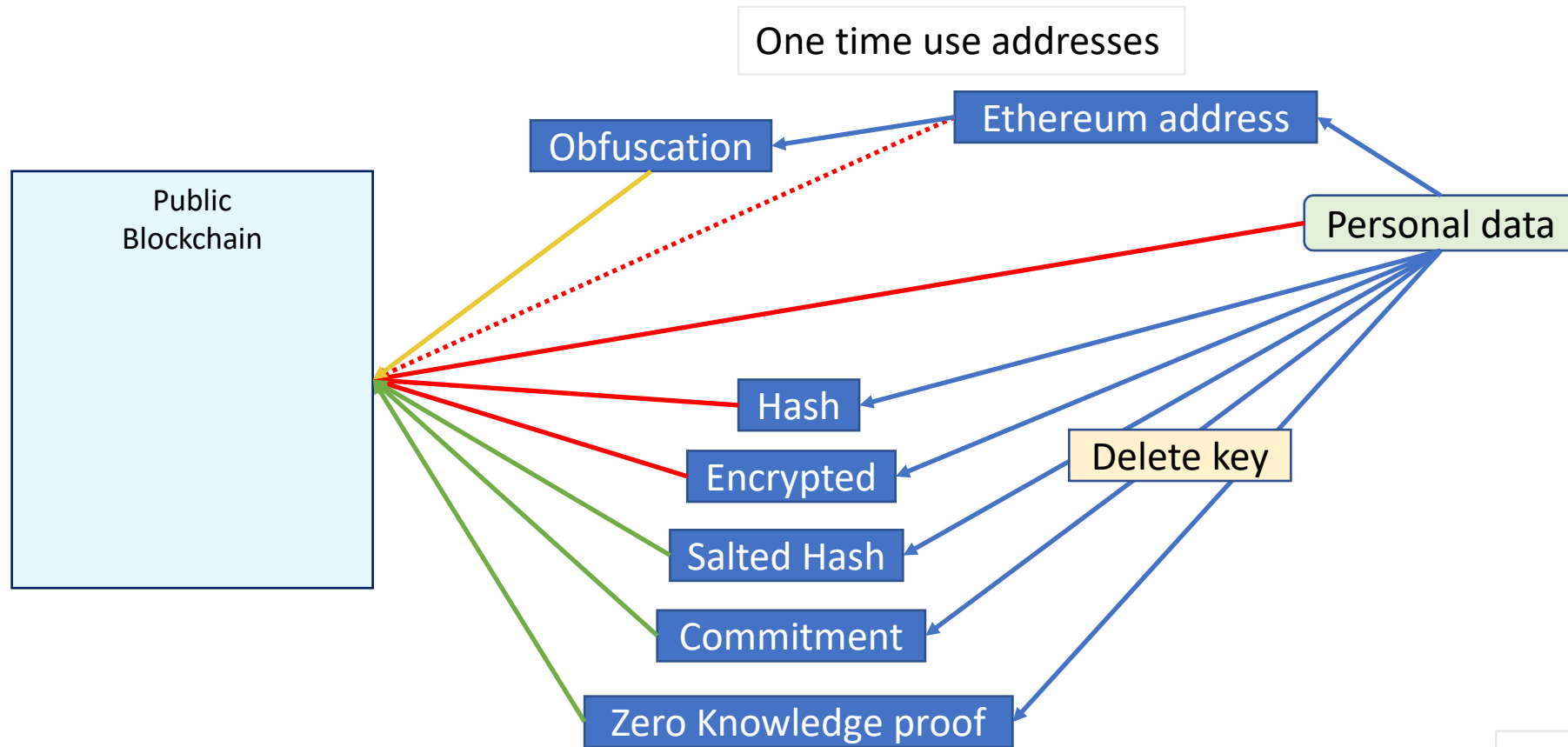
PD-12.4 Access control 2/2: Roles

```
role.sol x
1  // based on: https://docs.openzeppelin.com/contracts/3.x/access-control
2  // SPDX-License-Identifier: MIT
3  pragma solidity ^0.6.0;
4
5  import "@openzeppelin/contracts/access/AccessControl.sol";
6  import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
7
8  contract MyToken is ERC20, AccessControl {
9      bytes32 public constant MINTER_ROLE = keccak256("MINTER_ROLE");
10     bytes32 public constant BURNER_ROLE = keccak256("BURNER_ROLE");
11
12     constructor(address minter, address burner) public ERC20("MyToken", "TKN") {
13         _setupRole(MINTER_ROLE, minter);
14         _setupRole(BURNER_ROLE, burner);
15     }
16
17     function mint(address to, uint256 amount) public {
18         require(hasRole(MINTER_ROLE, msg.sender), "Caller is not a minter");
19         _mint(to, amount);
20     }
21
22     function burn(address from, uint256 amount) public {
23         require(hasRole(BURNER_ROLE, msg.sender), "Caller is not a burner");
24         _burn(from, amount);
25     }
26 }
```

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

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

(GDPR)



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

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

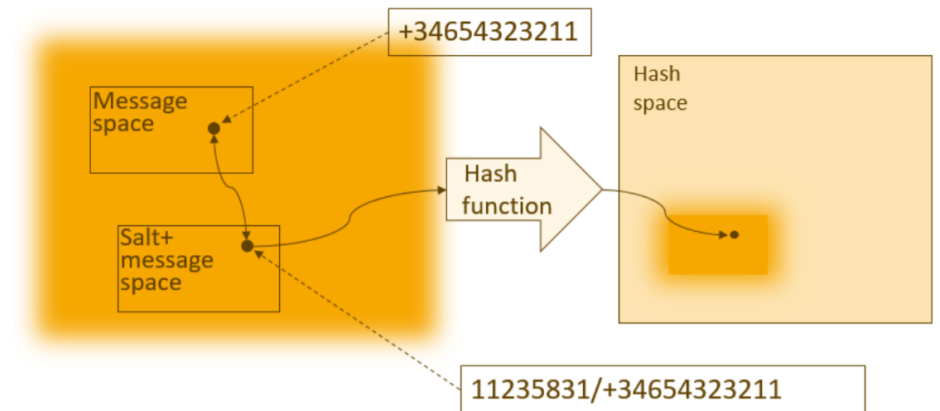
https://www.cravath.com/files/Uploads/Documents/Publications/3900063_1.pdf

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

encryption = pseudonymization
≠ anonymization

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



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

PD-12.6 Steel wallet

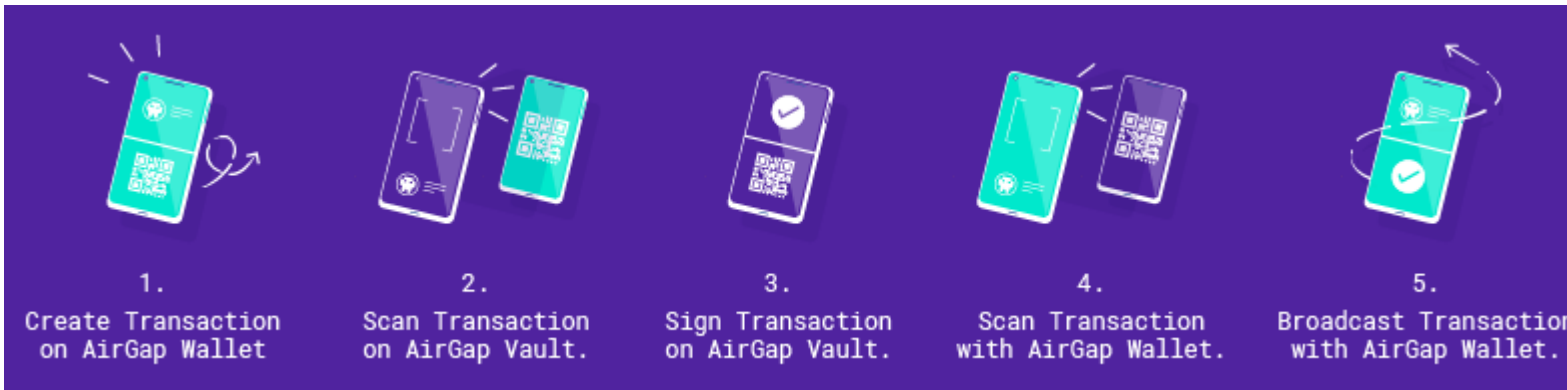


<https://cryptosteel.com>

<https://www.blockplate.com/>

PD-12.6 Offline / airgap

- Offline signing



SUMMARY OF EXISTING AIR-GAP COVERT CHANNELS

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

<https://airgap.it/>

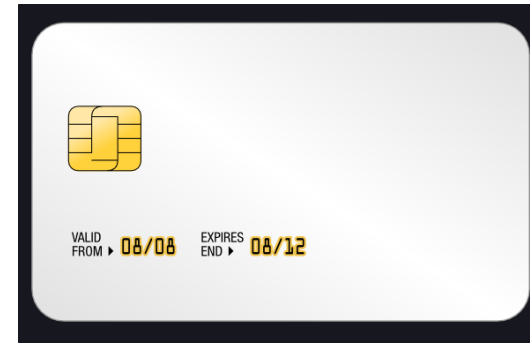
<https://medium.com/airgap-it/airgap-the-step-by-step-guide-c4c3d3fe9a05>

<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

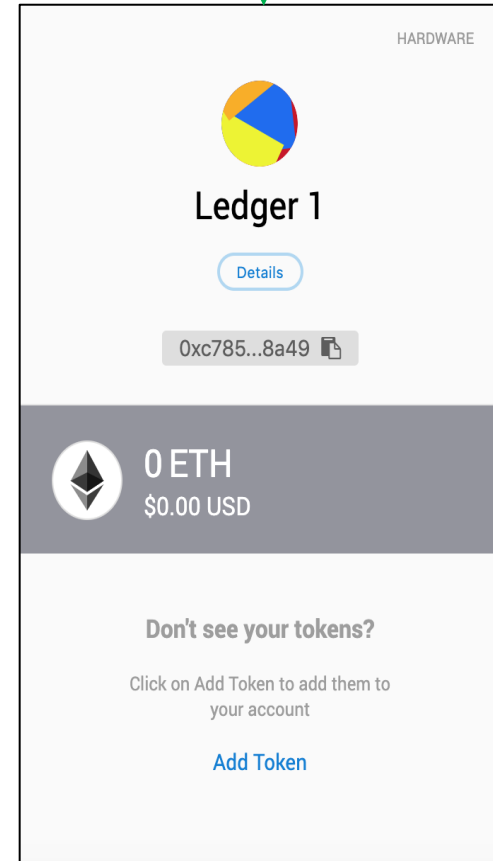
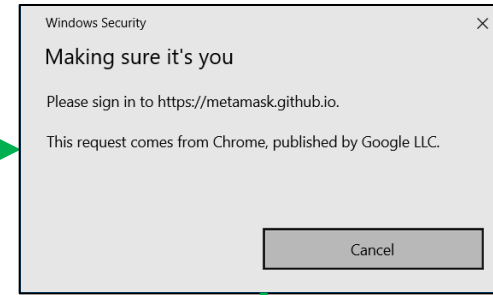
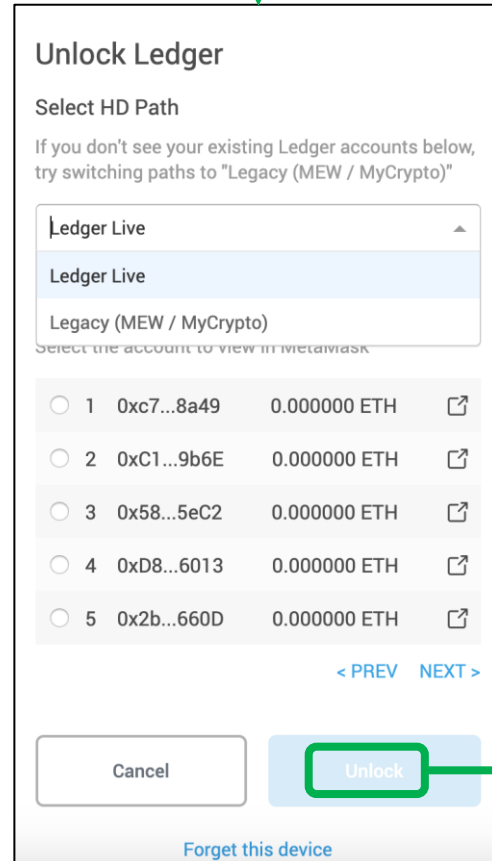
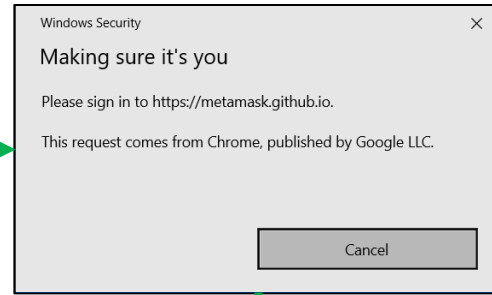
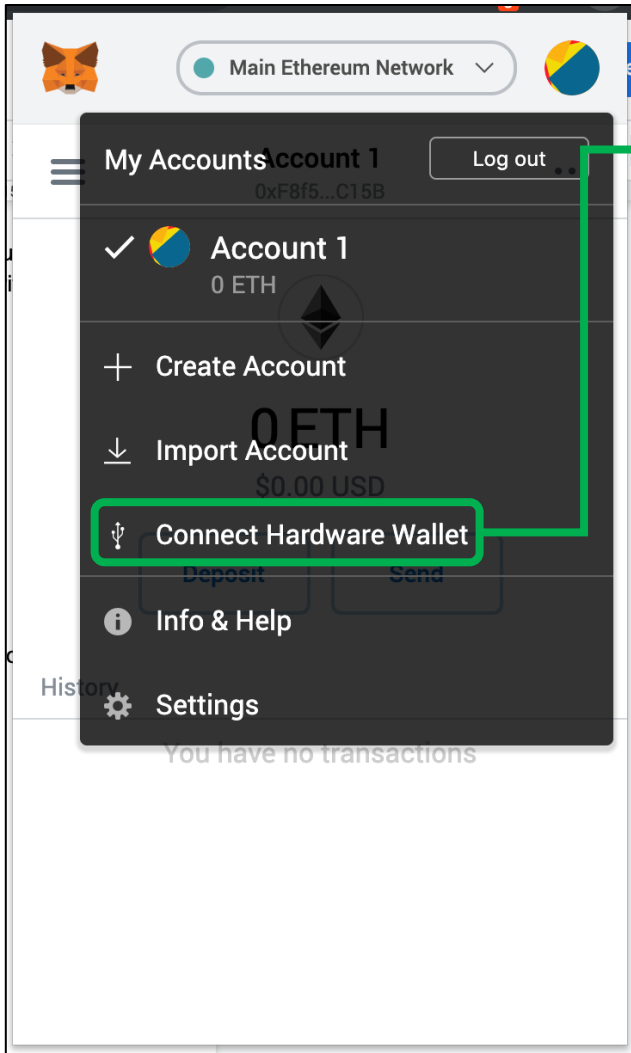
<https://wallet.fail/wallets/nanos/>

<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



PD-12.7 Ledger via Javascript (low level API)

```
low_level_ledger.js x
1  const TransportHid = require("@ledgerhq/hw-transport-node-hid").default;
2  const AppEth = require("@ledgerhq/hw-app-eth").default;
3
4  async function f() {
5      const transport = await TransportHid.create().catch(x=>console.log(`Error: ${x.message}`));
6      if (transport) {
7          console.log(`Connected to ${transport.deviceModel.id}`);
8          const eth = new AppEth(transport);
9          var res = await eth.getAppConfiguration();
10         console.log(`Version: ${res.version}`);
11         var keypair = await eth.getAddress("44'/60'/0'/0'/0").catch(x=>console.log(`Error: ${x.message}`));
12         if (keypair)
13             console.log(`Lowlevel address: ${keypair.address}`)
14     }
15 }
16 f();
```

> node low_level_ledger.js

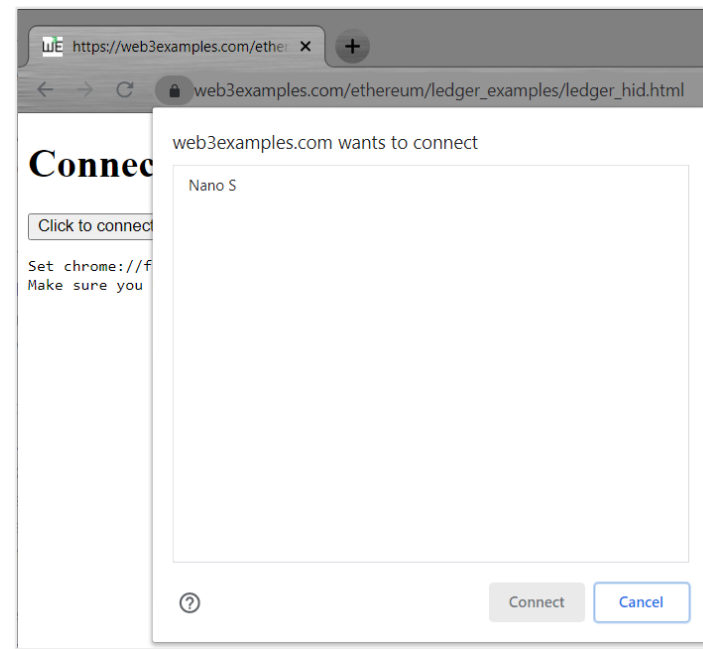
Connected to nanoS

Version: 1.6.2

Lowlevel address:0x.....

PD-12.7 Ledger via WebUSB

```
ledger_hid.html x
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <script src="webusb-browserify.js"></script>
5   </head>
6   <body>
7     <h1>Connect to Ledger via HID</h1>
8     <input type="button" value="Click to connect" onclick="f()">
9     <pre id="log" style="width:100%;height:200px"></pre>
10  <script>
11    function log(logstr) {
12      document.getElementById("log").innerHTML += logstr + "\n";
13    }
14    log("Set chrome://flags/#enable-experimental-web-platform-features");
15    log("Make sure you have the latest firmware on the ledger (at least 1.6.1)")
16    async function f() {
17      const transport = await TransportWebUSB.create().catch(x=>log(`Error: ${x.message}`));
18      if (transport) {
19        log(`Connected to ${transport.deviceModel.id}`);
20        const eth = new AppEth(transport);
21        var res = await eth.getAppConfiguration().catch(x=>log(`Error: ${x.message}`));
22        log(`Software version: ${res.version}`);
23        var keypair = await eth.getAddress("44'/60'/0'/0/0").catch(x=>log(`Error: ${x.message}`));
24        if (keypair)
25          log(`Lowlevel address: ${keypair.address}`);
26        const engine = new ProviderEngine();
27        getTransport = () => transport;
28        const rpcUrl = "https://cloudflare-eth.com"; // "https://ropsten.infura.io/";
29        const networkId = 1;
30        const ledger = createLedgerSubprovider(getTransport, {
31          networkId,
32          accountsLength: 1, // nr of accounts retrieved
33        });
34        engine.addProvider(ledger);
35        engine.addProvider(new RpcSubprovider({ rpcUrl }));
36        engine.start(); // start polling for blocks
37        const web3 = new Web3(engine);
38        var acts = await web3.eth.getAccounts().catch(x=>console.log(`Error: ${x.message}`));
39        if (acts)
40          log(`Via Web3 address: ${acts[0]}`); // accounts of ledger
41      }
42    }
43  </script>
```



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
```

```
generate_mnemonic.js x  
1  const bip39 = require('bip39');  
2  const mnemonic = bip39.generateMnemonic();  
3  console.log(mnemonic);
```

```
get_address_from_mnemonic.js x  
1  const Web3 = require('web3');  
2  const web3 = new Web3();  
3  const bip39 = require('bip39');  
4  const HDKey = require('hdkey');  
5  const mnemonic = "post soda ozone trash forget egg regret wink length minor winner broken";  
6  console.log(`Start mnemonic: ${mnemonic}`);  
7  
8  const seed = bip39.mnemonicToSeedSync(mnemonic);  
9  const hdWallet = HDKey.fromMasterSeed(seed);  
10 const root = hdWallet.derive("m/44'/60'/0'/0/0");  
11 const privateKey = "0x" + root.privateKey.toString('hex');  
12 console.log(`Private key: .....${privateKey}`);  
13  
14 var account = web3.eth.accounts.privateKeyToAccount(privateKey);  
15 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

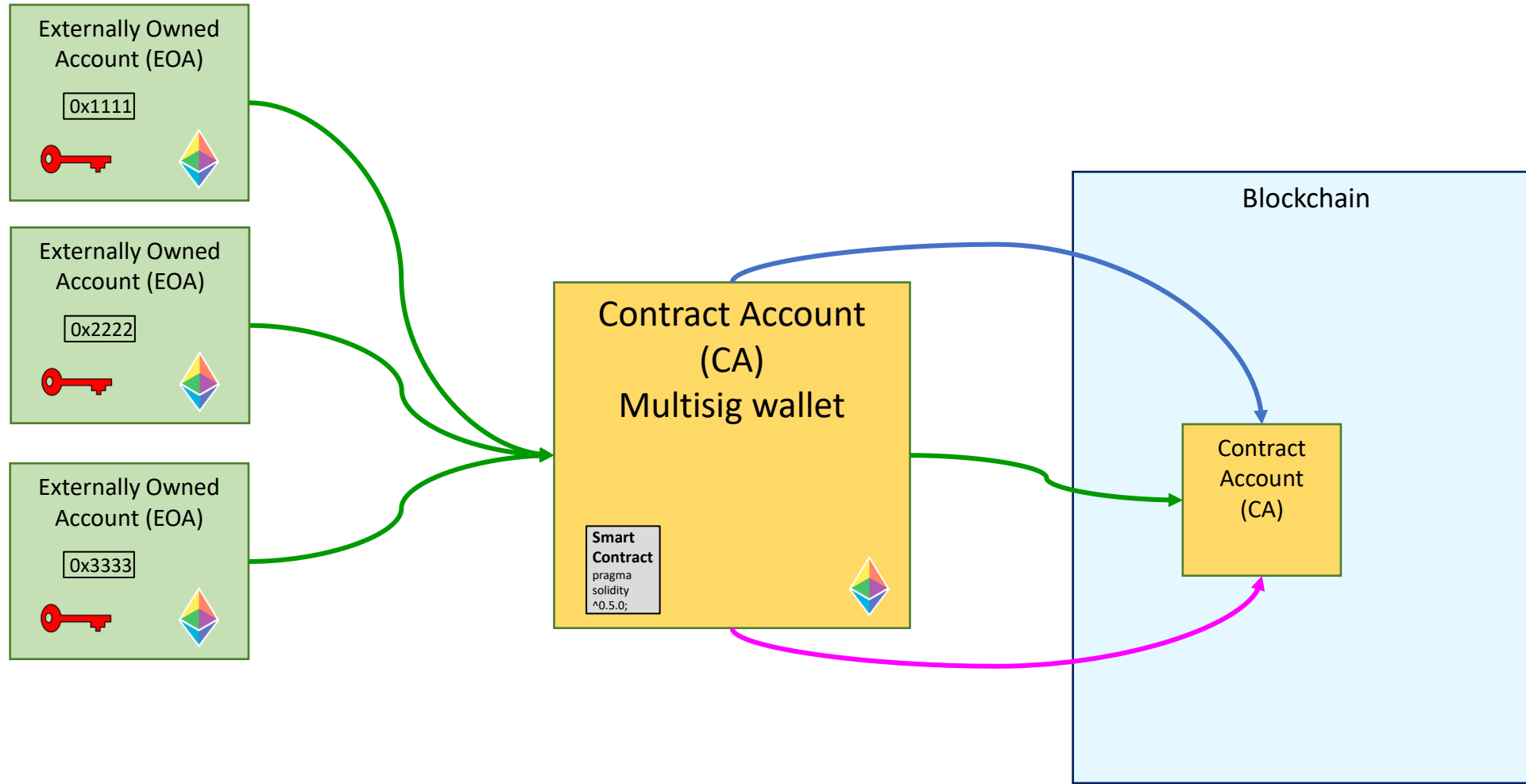
https://github.com/web3examples/ethereum/tree/master/wallet_examples/get_address_from_mnemonic.js

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

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

The screenshot shows the Gnosis Safe Multisig interface in a browser. The address bar displays the URL: rinkeby.gnosis-safe.io/safes/0x922B85A19E327762BA2877d35A090e8B0D4B0dFc/balances. The interface includes a header with the Gnosis Safe Multisig logo, a 'Safes' dropdown menu showing '1' safe, and a 'Rinkeby' network selector. A Metamask [RINKEBY] wallet is connected with the address 0xEA...a1c5. The main content area features a profile for 'Web3examples' with the address 0x922B85A19E327762BA2877d35A090e8B0D4B0dFc and 'Send' and 'Receive' buttons. Below this are tabs for 'BALANCES', 'TRANSACTIONS', and 'SETTINGS'. The 'BALANCES' tab is active, showing a table of assets with columns for 'ASSET', 'BALANCE', and 'VALUE'. The table lists Ether and W3ET. A 'Manage List' link and a currency selector set to 'USD' are also visible. At the bottom right, there is a chat icon.

Web3examples
0x922B85A19E327762BA2877d35A090e8B0D4B0dFc

[Send](#) [Receive](#)

BALANCES | TRANSACTIONS | SETTINGS

USD [Manage List](#)

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

Rows per page: 10 | 1-2 of 2 | < >

PD-12.9 Multisig Basics

```
multisigprep.sol x
1  // SPDX-License-Identifier: MIT
2  pragma solidity ^0.7.0;
3
4  contract MultisigPrep {
5      address public _savedest;
6      uint public _savevalue;
7      bytes public _savedata;
8      string public _stored;
9      event str(string);
10
11     function Store(string calldata message) external returns (string memory) {
12         _stored = message;
13         return _stored;
14     }
15     function Prepare(address destination, uint value, bytes calldata data) external {
16         _savedest = destination;
17         _savevalue = value;
18         _savedata = data;
19     }
20     function Execute() external returns (bytes memory) {
21         require(_savedest != address(0), "Not prepared");
22         (bool success, bytes memory res) = _savedest.call{value: _savevalue}(_savedata);
23         require(success, "Failed to execute transaction");
24         _savedest = address(0);
25         return res;
26     }
27     function TestMultisig() public {
28         this.Prepare(address(this), 0, abi.encodeWithSignature('Store(string)', 'Hello'));
29         bytes memory res = this.Execute();
30         string memory resstring = abi.decode(res, (string));
31         emit str(resstring);
32     }
33 }
```


PD-12.10 Security tools: Remix: Static analysis

The screenshot displays the Remix IDE interface. On the left, the 'SOLIDITY STATIC ANALYSIS' sidebar is open, showing various security and gas-related settings. The main area shows the Solidity code for 'VeryBasicNFT.sol'. On the right, the 'Results' pane displays several orange warning boxes.

SOLIDITY STATIC ANALYSIS Settings:

- 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:

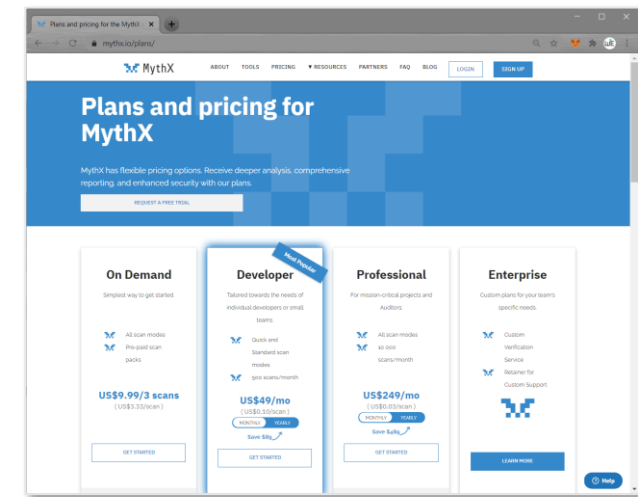
- Gas requirement of function `Token_erc20.name()` high: infinite. If the gas requirement of a function is higher than the block gas limit, it cannot be executed. Please avoid loops in your functions or actions that modify large areas of storage (this includes clearing or copying arrays in storage)
-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

Solidity Code (VeryBasicNFT.sol):

```
1 // Adapted from https://github.com/OpenZeppelin/openzeppelin-contracts/tree/master/contracts/token/ERC20
2 // https://raw.githubusercontent.com/web3examples/ethereum/master/token_examples/VeryBasicToken.sol
3
4 pragma solidity ^0.5.12;
5
6 contract Token_erc20 {
7     string private _name;
8     string private _symbol;
9     uint8 private _decimals;
10    mapping (address => uint256) private _balances;
11    uint256 private _totalSupply;
12
13    event Transfer(address indexed from, address indexed to, uint256 value);
14
15    constructor (string memory name, string memory symbol, uint8 decimals) public {
16        _name = name;
17        _symbol = symbol;
18        _decimals = decimals;
19        _mint(msg.sender, 10000 * (10 ** uint256(_decimals)));
20    }
21
22    function name() public view returns (string memory) {
23        return _name;
24    }
25
26    function symbol() public view returns (string memory) {
27        return _symbol;
28    }
29
30    function decimals() public view returns (uint8) {
31        return _decimals;
32    }
33
34    function add(uint256 a, uint256 b) internal pure returns (uint256) {
35        uint256 c = a + b;
36        require(c >= a, "SafeMath: addition overflow");
37        return c;
38    }
39
40    function sub(uint256 a, uint256 b) internal pure returns (uint256) {
41        require(b <= a, "SafeMath: subtraction overflow");
42        uint256 c = a - b;
43        return c;
44    }
45
46    function totalSupply() public view returns (uint256) {
47        return _totalSupply;
48    }
49
50    function balanceOf(address account) public view returns (uint256) {
51        return _balances[account];
52    }
53
54    function transfer(address recipient, uint256 amount) public returns (bool) {
55        _transfer(msg.sender, recipient, amount);
56        return true;
57    }
58
59    function _transfer(address sender, address recipient, uint256 amount) private {
60        require(sender != recipient, "SafeMath: self-transfer");
61        require(amount > 0, "SafeMath: amount too low");
62        require(_balances[sender] >= amount, "SafeMath: balance too low");
63        _balances[sender] = _balances[sender] - amount;
64        _balances[recipient] = _balances[recipient] + amount;
65        emit Transfer(sender, recipient, amount);
66    }
67
68    function _mint(address to, uint256 amount) private {
69        require(to != msg.sender, "SafeMath: self-mint");
70        _totalSupply = _totalSupply + amount;
71        _balances[to] = _balances[to] + amount;
72    }
73
74    function _burn(address account, uint256 amount) private {
75        require(_balances[account] >= amount, "SafeMath: balance too low");
76        _balances[account] = _balances[account] - amount;
77        emit Transfer(account, address(0), amount);
78    }
79
80    function _approve(address owner, address spender, uint256 amount) private {
81        require(owner != spender, "SafeMath: self-approve");
82        require(_balances[owner] >= amount, "SafeMath: balance too low");
83        _balances[owner] = _balances[owner] - amount;
84        _balances[spender] = _balances[spender] + amount;
85        emit Approval(owner, spender, amount);
86    }
87
88    function _transferFrom(address sender, address recipient, uint256 amount) private {
89        require(sender != recipient, "SafeMath: self-transfer");
90        require(amount > 0, "SafeMath: amount too low");
91        require(_balances[sender] >= amount, "SafeMath: balance too low");
92        _balances[sender] = _balances[sender] - amount;
93        _balances[recipient] = _balances[recipient] + amount;
94        emit Transfer(sender, recipient, amount);
95    }
96
97    function _approveFrom(address owner, address spender, uint256 amount) private {
98        require(owner != spender, "SafeMath: self-approve");
99        require(_balances[owner] >= amount, "SafeMath: balance too low");
100    _balances[owner] = _balances[owner] - amount;
101    _balances[spender] = _balances[spender] + amount;
102    emit Approval(owner, spender, amount);
103    }
104
105    function _burnFrom(address account, uint256 amount) private {
106        require(_balances[account] >= amount, "SafeMath: balance too low");
107        _balances[account] = _balances[account] - amount;
108        emit Transfer(account, address(0), amount);
109    }
110
111    function _transferFrom(address sender, address recipient, uint256 amount, bool isApproval) private {
112        require(sender != recipient, "SafeMath: self-transfer");
113        require(amount > 0, "SafeMath: amount too low");
114        require(_balances[sender] >= amount, "SafeMath: balance too low");
115        _balances[sender] = _balances[sender] - amount;
116        _balances[recipient] = _balances[recipient] + amount;
117        emit Transfer(sender, recipient, amount);
118        if (isApproval) {
119            emit Approval(sender, recipient, amount);
120        }
121    }
122
123    function _approveFrom(address owner, address spender, uint256 amount, bool isApproval) private {
124        require(owner != spender, "SafeMath: self-approve");
125        require(_balances[owner] >= amount, "SafeMath: balance too low");
126        _balances[owner] = _balances[owner] - amount;
127        _balances[spender] = _balances[spender] + amount;
128        emit Approval(owner, spender, amount);
129        if (isApproval) {
130            emit ApprovalFrom(owner, spender, amount);
131        }
132    }
133
134    function _burnFrom(address account, uint256 amount, bool isApproval) private {
135        require(_balances[account] >= amount, "SafeMath: balance too low");
136        _balances[account] = _balances[account] - amount;
137        emit Transfer(account, address(0), amount);
138        if (isApproval) {
139            emit BurnFrom(account, amount);
140        }
141    }
142
143    function _transferFrom(address sender, address recipient, uint256 amount, bool isApproval, bool isBurnFrom) private {
144        require(sender != recipient, "SafeMath: self-transfer");
145        require(amount > 0, "SafeMath: amount too low");
146        require(_balances[sender] >= amount, "SafeMath: balance too low");
147        _balances[sender] = _balances[sender] - amount;
148        _balances[recipient] = _balances[recipient] + amount;
149        emit Transfer(sender, recipient, amount);
150        if (isApproval) {
151            emit Approval(sender, recipient, amount);
152        }
153        if (isBurnFrom) {
154            emit BurnFrom(sender, amount);
155        }
156    }
157
158    function _approveFrom(address owner, address spender, uint256 amount, bool isApproval, bool isBurnFrom) private {
159        require(owner != spender, "SafeMath: self-approve");
160        require(_balances[owner] >= amount, "SafeMath: balance too low");
161        _balances[owner] = _balances[owner] - amount;
162        _balances[spender] = _balances[spender] + amount;
163        emit Approval(owner, spender, amount);
164        if (isApproval) {
165            emit ApprovalFrom(owner, spender, amount);
166        }
167        if (isBurnFrom) {
168            emit BurnFrom(owner, amount);
169        }
170    }
171
172    function _burnFrom(address account, uint256 amount, bool isApproval, bool isBurnFrom) private {
173        require(_balances[account] >= amount, "SafeMath: balance too low");
174        _balances[account] = _balances[account] - amount;
175        emit Transfer(account, address(0), amount);
176        if (isApproval) {
177            emit BurnFrom(account, amount);
178        }
179        if (isBurnFrom) {
180            emit BurnFrom(account, amount);
181        }
182    }
183
184    function _transferFrom(address sender, address recipient, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom) private {
185        require(sender != recipient, "SafeMath: self-transfer");
186        require(amount > 0, "SafeMath: amount too low");
187        require(_balances[sender] >= amount, "SafeMath: balance too low");
188        _balances[sender] = _balances[sender] - amount;
189        _balances[recipient] = _balances[recipient] + amount;
190        emit Transfer(sender, recipient, amount);
191        if (isApproval) {
192            emit Approval(sender, recipient, amount);
193        }
194        if (isBurnFrom) {
195            emit BurnFrom(sender, amount);
196        }
197        if (isBurnFromFrom) {
198            emit BurnFromFrom(sender, amount);
199        }
200    }
201
202    function _approveFrom(address owner, address spender, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom) private {
203        require(owner != spender, "SafeMath: self-approve");
204        require(_balances[owner] >= amount, "SafeMath: balance too low");
205        _balances[owner] = _balances[owner] - amount;
206        _balances[spender] = _balances[spender] + amount;
207        emit Approval(owner, spender, amount);
208        if (isApproval) {
209            emit ApprovalFrom(owner, spender, amount);
210        }
211        if (isBurnFrom) {
212            emit BurnFrom(owner, amount);
213        }
214        if (isBurnFromFrom) {
215            emit BurnFromFrom(owner, amount);
216        }
217    }
218
219    function _burnFrom(address account, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom) private {
220        require(_balances[account] >= amount, "SafeMath: balance too low");
221        _balances[account] = _balances[account] - amount;
222        emit Transfer(account, address(0), amount);
223        if (isApproval) {
224            emit BurnFrom(account, amount);
225        }
226        if (isBurnFrom) {
227            emit BurnFrom(account, amount);
228        }
229        if (isBurnFromFrom) {
230            emit BurnFromFrom(account, amount);
231        }
232    }
233
234    function _transferFrom(address sender, address recipient, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom) private {
235        require(sender != recipient, "SafeMath: self-transfer");
236        require(amount > 0, "SafeMath: amount too low");
237        require(_balances[sender] >= amount, "SafeMath: balance too low");
238        _balances[sender] = _balances[sender] - amount;
239        _balances[recipient] = _balances[recipient] + amount;
240        emit Transfer(sender, recipient, amount);
241        if (isApproval) {
242            emit Approval(sender, recipient, amount);
243        }
244        if (isBurnFrom) {
245            emit BurnFrom(sender, amount);
246        }
247        if (isBurnFromFrom) {
248            emit BurnFromFrom(sender, amount);
249        }
250        if (isBurnFromFromFrom) {
251            emit BurnFromFromFrom(sender, amount);
252        }
253    }
254
255    function _approveFrom(address owner, address spender, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom) private {
256        require(owner != spender, "SafeMath: self-approve");
257        require(_balances[owner] >= amount, "SafeMath: balance too low");
258        _balances[owner] = _balances[owner] - amount;
259        _balances[spender] = _balances[spender] + amount;
260        emit Approval(owner, spender, amount);
261        if (isApproval) {
262            emit ApprovalFrom(owner, spender, amount);
263        }
264        if (isBurnFrom) {
265            emit BurnFrom(owner, amount);
266        }
267        if (isBurnFromFrom) {
268            emit BurnFromFrom(owner, amount);
269        }
270        if (isBurnFromFromFrom) {
271            emit BurnFromFromFrom(owner, amount);
272        }
273    }
274
275    function _burnFrom(address account, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom) private {
276        require(_balances[account] >= amount, "SafeMath: balance too low");
277        _balances[account] = _balances[account] - amount;
278        emit Transfer(account, address(0), amount);
279        if (isApproval) {
280            emit BurnFrom(account, amount);
281        }
282        if (isBurnFrom) {
283            emit BurnFrom(account, amount);
284        }
285        if (isBurnFromFrom) {
286            emit BurnFromFrom(account, amount);
287        }
288        if (isBurnFromFromFrom) {
289            emit BurnFromFromFrom(account, amount);
290        }
291    }
292
293    function _transferFrom(address sender, address recipient, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom) private {
294        require(sender != recipient, "SafeMath: self-transfer");
295        require(amount > 0, "SafeMath: amount too low");
296        require(_balances[sender] >= amount, "SafeMath: balance too low");
297        _balances[sender] = _balances[sender] - amount;
298        _balances[recipient] = _balances[recipient] + amount;
299        emit Transfer(sender, recipient, amount);
300        if (isApproval) {
301            emit Approval(sender, recipient, amount);
302        }
303        if (isBurnFrom) {
304            emit BurnFrom(sender, amount);
305        }
306        if (isBurnFromFrom) {
307            emit BurnFromFrom(sender, amount);
308        }
309        if (isBurnFromFromFrom) {
310            emit BurnFromFromFrom(sender, amount);
311        }
312        if (isBurnFromFromFromFrom) {
313            emit BurnFromFromFromFrom(sender, amount);
314        }
315    }
316
317    function _approveFrom(address owner, address spender, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom) private {
318        require(owner != spender, "SafeMath: self-approve");
319        require(_balances[owner] >= amount, "SafeMath: balance too low");
320        _balances[owner] = _balances[owner] - amount;
321        _balances[spender] = _balances[spender] + amount;
322        emit Approval(owner, spender, amount);
323        if (isApproval) {
324            emit ApprovalFrom(owner, spender, amount);
325        }
326        if (isBurnFrom) {
327            emit BurnFrom(owner, amount);
328        }
329        if (isBurnFromFrom) {
330            emit BurnFromFrom(owner, amount);
331        }
332        if (isBurnFromFromFrom) {
333            emit BurnFromFromFrom(owner, amount);
334        }
335        if (isBurnFromFromFromFrom) {
336            emit BurnFromFromFromFrom(owner, amount);
337        }
338    }
339
340    function _burnFrom(address account, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom) private {
341        require(_balances[account] >= amount, "SafeMath: balance too low");
342        _balances[account] = _balances[account] - amount;
343        emit Transfer(account, address(0), amount);
344        if (isApproval) {
345            emit BurnFrom(account, amount);
346        }
347        if (isBurnFrom) {
348            emit BurnFrom(account, amount);
349        }
350        if (isBurnFromFrom) {
351            emit BurnFromFrom(account, amount);
352        }
353        if (isBurnFromFromFrom) {
354            emit BurnFromFromFrom(account, amount);
355        }
356        if (isBurnFromFromFromFrom) {
357            emit BurnFromFromFromFrom(account, amount);
358        }
359    }
360
361    function _transferFrom(address sender, address recipient, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom) private {
362        require(sender != recipient, "SafeMath: self-transfer");
363        require(amount > 0, "SafeMath: amount too low");
364        require(_balances[sender] >= amount, "SafeMath: balance too low");
365        _balances[sender] = _balances[sender] - amount;
366        _balances[recipient] = _balances[recipient] + amount;
367        emit Transfer(sender, recipient, amount);
368        if (isApproval) {
369            emit Approval(sender, recipient, amount);
370        }
371        if (isBurnFrom) {
372            emit BurnFrom(sender, amount);
373        }
374        if (isBurnFromFrom) {
375            emit BurnFromFrom(sender, amount);
376        }
377        if (isBurnFromFromFrom) {
378            emit BurnFromFromFrom(sender, amount);
379        }
380        if (isBurnFromFromFromFrom) {
381            emit BurnFromFromFromFrom(sender, amount);
382        }
383        if (isBurnFromFromFromFromFrom) {
384            emit BurnFromFromFromFromFrom(sender, amount);
385        }
386    }
387
388    function _approveFrom(address owner, address spender, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom) private {
389        require(owner != spender, "SafeMath: self-approve");
390        require(_balances[owner] >= amount, "SafeMath: balance too low");
391        _balances[owner] = _balances[owner] - amount;
392        _balances[spender] = _balances[spender] + amount;
393        emit Approval(owner, spender, amount);
394        if (isApproval) {
395            emit ApprovalFrom(owner, spender, amount);
396        }
397        if (isBurnFrom) {
398            emit BurnFrom(owner, amount);
399        }
400        if (isBurnFromFrom) {
401            emit BurnFromFrom(owner, amount);
402        }
403        if (isBurnFromFromFrom) {
404            emit BurnFromFromFrom(owner, amount);
405        }
406        if (isBurnFromFromFromFrom) {
407            emit BurnFromFromFromFrom(owner, amount);
408        }
409        if (isBurnFromFromFromFromFrom) {
410            emit BurnFromFromFromFromFrom(owner, amount);
411        }
412    }
413
414    function _burnFrom(address account, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom) private {
415        require(_balances[account] >= amount, "SafeMath: balance too low");
416        _balances[account] = _balances[account] - amount;
417        emit Transfer(account, address(0), amount);
418        if (isApproval) {
419            emit BurnFrom(account, amount);
420        }
421        if (isBurnFrom) {
422            emit BurnFrom(account, amount);
423        }
424        if (isBurnFromFrom) {
425            emit BurnFromFrom(account, amount);
426        }
427        if (isBurnFromFromFrom) {
428            emit BurnFromFromFrom(account, amount);
429        }
430        if (isBurnFromFromFromFrom) {
431            emit BurnFromFromFromFrom(account, amount);
432        }
433        if (isBurnFromFromFromFromFrom) {
434            emit BurnFromFromFromFromFrom(account, amount);
435        }
436    }
437
438    function _transferFrom(address sender, address recipient, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom, bool isBurnFromFromFromFromFromFrom) private {
439        require(sender != recipient, "SafeMath: self-transfer");
440        require(amount > 0, "SafeMath: amount too low");
441        require(_balances[sender] >= amount, "SafeMath: balance too low");
442        _balances[sender] = _balances[sender] - amount;
443        _balances[recipient] = _balances[recipient] + amount;
444        emit Transfer(sender, recipient, amount);
445        if (isApproval) {
446            emit Approval(sender, recipient, amount);
447        }
448        if (isBurnFrom) {
449            emit BurnFrom(sender, amount);
450        }
451        if (isBurnFromFrom) {
452            emit BurnFromFrom(sender, amount);
453        }
454        if (isBurnFromFromFrom) {
455            emit BurnFromFromFrom(sender, amount);
456        }
457        if (isBurnFromFromFromFrom) {
458            emit BurnFromFromFromFrom(sender, amount);
459        }
460        if (isBurnFromFromFromFromFrom) {
461            emit BurnFromFromFromFromFrom(sender, amount);
462        }
463        if (isBurnFromFromFromFromFromFrom) {
464            emit BurnFromFromFromFromFromFrom(sender, amount);
465        }
466    }
467
468    function _approveFrom(address owner, address spender, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom, bool isBurnFromFromFromFromFromFrom) private {
469        require(owner != spender, "SafeMath: self-approve");
470        require(_balances[owner] >= amount, "SafeMath: balance too low");
471        _balances[owner] = _balances[owner] - amount;
472        _balances[spender] = _balances[spender] + amount;
473        emit Approval(owner, spender, amount);
474        if (isApproval) {
475            emit ApprovalFrom(owner, spender, amount);
476        }
477        if (isBurnFrom) {
478            emit BurnFrom(owner, amount);
479        }
480        if (isBurnFromFrom) {
481            emit BurnFromFrom(owner, amount);
482        }
483        if (isBurnFromFromFrom) {
484            emit BurnFromFromFrom(owner, amount);
485        }
486        if (isBurnFromFromFromFrom) {
487            emit BurnFromFromFromFrom(owner, amount);
488        }
489        if (isBurnFromFromFromFromFrom) {
490            emit BurnFromFromFromFromFrom(owner, amount);
491        }
492        if (isBurnFromFromFromFromFromFrom) {
493            emit BurnFromFromFromFromFromFrom(owner, amount);
494        }
495    }
496
497    function _burnFrom(address account, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom, bool isBurnFromFromFromFromFromFrom) private {
498        require(_balances[account] >= amount, "SafeMath: balance too low");
499        _balances[account] = _balances[account] - amount;
500        emit Transfer(account, address(0), amount);
501        if (isApproval) {
502            emit BurnFrom(account, amount);
503        }
504        if (isBurnFrom) {
505            emit BurnFrom(account, amount);
506        }
507        if (isBurnFromFrom) {
508            emit BurnFromFrom(account, amount);
509        }
510        if (isBurnFromFromFrom) {
511            emit BurnFromFromFrom(account, amount);
512        }
513        if (isBurnFromFromFromFrom) {
514            emit BurnFromFromFromFrom(account, amount);
515        }
516        if (isBurnFromFromFromFromFrom) {
517            emit BurnFromFromFromFromFrom(account, amount);
518        }
519        if (isBurnFromFromFromFromFromFrom) {
520            emit BurnFromFromFromFromFromFrom(account, amount);
521        }
522    }
523
524    function _transferFrom(address sender, address recipient, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom, bool isBurnFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFrom) private {
525        require(sender != recipient, "SafeMath: self-transfer");
526        require(amount > 0, "SafeMath: amount too low");
527        require(_balances[sender] >= amount, "SafeMath: balance too low");
528        _balances[sender] = _balances[sender] - amount;
529        _balances[recipient] = _balances[recipient] + amount;
530        emit Transfer(sender, recipient, amount);
531        if (isApproval) {
532            emit Approval(sender, recipient, amount);
533        }
534        if (isBurnFrom) {
535            emit BurnFrom(sender, amount);
536        }
537        if (isBurnFromFrom) {
538            emit BurnFromFrom(sender, amount);
539        }
540        if (isBurnFromFromFrom) {
541            emit BurnFromFromFrom(sender, amount);
542        }
543        if (isBurnFromFromFromFrom) {
544            emit BurnFromFromFromFrom(sender, amount);
545        }
546        if (isBurnFromFromFromFromFrom) {
547            emit BurnFromFromFromFromFrom(sender, amount);
548        }
549        if (isBurnFromFromFromFromFromFrom) {
550            emit BurnFromFromFromFromFromFrom(sender, amount);
551        }
552        if (isBurnFromFromFromFromFromFromFrom) {
553            emit BurnFromFromFromFromFromFromFrom(sender, amount);
554        }
555    }
556
557    function _approveFrom(address owner, address spender, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom, bool isBurnFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFrom) private {
558        require(owner != spender, "SafeMath: self-approve");
559        require(_balances[owner] >= amount, "SafeMath: balance too low");
560        _balances[owner] = _balances[owner] - amount;
561        _balances[spender] = _balances[spender] + amount;
562        emit Approval(owner, spender, amount);
563        if (isApproval) {
564            emit ApprovalFrom(owner, spender, amount);
565        }
566        if (isBurnFrom) {
567            emit BurnFrom(owner, amount);
568        }
569        if (isBurnFromFrom) {
570            emit BurnFromFrom(owner, amount);
571        }
572        if (isBurnFromFromFrom) {
573            emit BurnFromFromFrom(owner, amount);
574        }
575        if (isBurnFromFromFromFrom) {
576            emit BurnFromFromFromFrom(owner, amount);
577        }
578        if (isBurnFromFromFromFromFrom) {
579            emit BurnFromFromFromFromFrom(owner, amount);
580        }
581        if (isBurnFromFromFromFromFromFrom) {
582            emit BurnFromFromFromFromFromFrom(owner, amount);
583        }
584        if (isBurnFromFromFromFromFromFromFrom) {
585            emit BurnFromFromFromFromFromFromFrom(owner, amount);
586        }
587    }
588
589    function _burnFrom(address account, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom, bool isBurnFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFrom) private {
590        require(_balances[account] >= amount, "SafeMath: balance too low");
591        _balances[account] = _balances[account] - amount;
592        emit Transfer(account, address(0), amount);
593        if (isApproval) {
594            emit BurnFrom(account, amount);
595        }
596        if (isBurnFrom) {
597            emit BurnFrom(account, amount);
598        }
599        if (isBurnFromFrom) {
600            emit BurnFromFrom(account, amount);
601        }
602        if (isBurnFromFromFrom) {
603            emit BurnFromFromFrom(account, amount);
604        }
605        if (isBurnFromFromFromFrom) {
606            emit BurnFromFromFromFrom(account, amount);
607        }
608        if (isBurnFromFromFromFromFrom) {
609            emit BurnFromFromFromFromFrom(account, amount);
610        }
611        if (isBurnFromFromFromFromFromFrom) {
612            emit BurnFromFromFromFromFromFrom(account, amount);
613        }
614        if (isBurnFromFromFromFromFromFromFrom) {
615            emit BurnFromFromFromFromFromFromFrom(account, amount);
616        }
617    }
618
619    function _transferFrom(address sender, address recipient, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom, bool isBurnFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFromFrom) private {
620        require(sender != recipient, "SafeMath: self-transfer");
621        require(amount > 0, "SafeMath: amount too low");
622        require(_balances[sender] >= amount, "SafeMath: balance too low");
623        _balances[sender] = _balances[sender] - amount;
624        _balances[recipient] = _balances[recipient] + amount;
625        emit Transfer(sender, recipient, amount);
626        if (isApproval) {
627            emit Approval(sender, recipient, amount);
628        }
629        if (isBurnFrom) {
630            emit BurnFrom(sender, amount);
631        }
632        if (isBurnFromFrom) {
633            emit BurnFromFrom(sender, amount);
634        }
635        if (isBurnFromFromFrom) {
636            emit BurnFromFromFrom(sender, amount);
637        }
638        if (isBurnFromFromFromFrom) {
639            emit BurnFromFromFromFrom(sender, amount);
640        }
641        if (isBurnFromFromFromFromFrom) {
642            emit BurnFromFromFromFromFrom(sender, amount);
643        }
644        if (isBurnFromFromFromFromFromFrom) {
645            emit BurnFromFromFromFromFromFrom(sender, amount);
646        }
647        if (isBurnFromFromFromFromFromFromFrom) {
648            emit BurnFromFromFromFromFromFromFrom(sender, amount);
649        }
650        if (isBurnFromFromFromFromFromFromFromFrom) {
651            emit BurnFromFromFromFromFromFromFromFrom(sender, amount);
652        }
653    }
654
655    function _approveFrom(address owner, address spender, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom, bool isBurnFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFromFrom) private {
656        require(owner != spender, "SafeMath: self-approve");
657        require(_balances[owner] >= amount, "SafeMath: balance too low");
658        _balances[owner] = _balances[owner] - amount;
659        _balances[spender] = _balances[spender] + amount;
660        emit Approval(owner, spender, amount);
661        if (isApproval) {
662            emit ApprovalFrom(owner, spender, amount);
663        }
664        if (isBurnFrom) {
665            emit BurnFrom(owner, amount);
666        }
667        if (isBurnFromFrom) {
668            emit BurnFromFrom(owner, amount);
669        }
670        if (isBurnFromFromFrom) {
671            emit BurnFromFromFrom(owner, amount);
672        }
673        if (isBurnFromFromFromFrom) {
674            emit BurnFromFromFromFrom(owner, amount);
675        }
676        if (isBurnFromFromFromFromFrom) {
677            emit BurnFromFromFromFromFrom(owner, amount);
678        }
679        if (isBurnFromFromFromFromFromFrom) {
680            emit BurnFromFromFromFromFromFrom(owner, amount);
681        }
682        if (isBurnFromFromFromFromFromFromFrom) {
683            emit BurnFromFromFromFromFromFromFrom(owner, amount);
684        }
685        if (isBurnFromFromFromFromFromFromFromFrom) {
686            emit BurnFromFromFromFromFromFromFromFrom(owner, amount);
687        }
688    }
689
690    function _burnFrom(address account, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom, bool isBurnFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFromFrom) private {
691        require(_balances[account] >= amount, "SafeMath: balance too low");
692        _balances[account] = _balances[account] - amount;
693        emit Transfer(account, address(0), amount);
694        if (isApproval) {
695            emit BurnFrom(account, amount);
696        }
697        if (isBurnFrom) {
698            emit BurnFrom(account, amount);
699        }
700        if (isBurnFromFrom) {
701            emit BurnFromFrom(account, amount);
702        }
703        if (isBurnFromFromFrom) {
704            emit BurnFromFromFrom(account, amount);
705        }
706        if (isBurnFromFromFromFrom) {
707            emit BurnFromFromFromFrom(account, amount);
708        }
709        if (isBurnFromFromFromFromFrom) {
710            emit BurnFromFromFromFromFrom(account, amount);
711        }
712        if (isBurnFromFromFromFromFromFrom) {
713            emit BurnFromFromFromFromFromFrom(account, amount);
714        }
715        if (isBurnFromFromFromFromFromFromFrom) {
716            emit BurnFromFromFromFromFromFromFrom(account, amount);
717        }
718        if (isBurnFromFromFromFromFromFromFromFrom) {
719            emit BurnFromFromFromFromFromFromFromFrom(account, amount);
720        }
721    }
722
723    function _transferFrom(address sender, address recipient, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom, bool isBurnFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFromFromFrom) private {
724        require(sender != recipient, "SafeMath: self-transfer");
725        require(amount > 0, "SafeMath: amount too low");
726        require(_balances[sender] >= amount, "SafeMath: balance too low");
727        _balances[sender] = _balances[sender] - amount;
728        _balances[recipient] = _balances[recipient] + amount;
729        emit Transfer(sender, recipient, amount);
730        if (isApproval) {
731            emit Approval(sender, recipient, amount);
732        }
733        if (isBurnFrom) {
734            emit BurnFrom(sender, amount);
735        }
736        if (isBurnFromFrom) {
737            emit BurnFromFrom(sender, amount);
738        }
739        if (isBurnFromFromFrom) {
740            emit BurnFromFromFrom(sender, amount);
741        }
742        if (isBurnFromFromFromFrom) {
743            emit BurnFromFromFromFrom(sender, amount);
744        }
745        if (isBurnFromFromFromFromFrom) {
746            emit BurnFromFromFromFromFrom(sender, amount);
747        }
748        if (isBurnFromFromFromFromFromFrom) {
749            emit BurnFromFromFromFromFromFrom(sender, amount);
750        }
751        if (isBurnFromFromFromFromFromFromFrom) {
752            emit BurnFromFromFromFromFromFromFrom(sender, amount);
753        }
754        if (isBurnFromFromFromFromFromFromFromFrom) {
755            emit BurnFromFromFromFromFromFromFromFrom(sender, amount);
756        }
757        if (isBurnFromFromFromFromFromFromFromFromFrom) {
758            emit BurnFromFromFromFromFromFromFromFromFrom(sender, amount);
759        }
760    }
761
762    function _approveFrom(address owner, address spender, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom, bool isBurnFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFromFromFrom) private {
763        require(owner != spender, "SafeMath: self-approve");
764        require(_balances[owner] >= amount, "SafeMath: balance too low");
765        _balances[owner] = _balances[owner] - amount;
766        _balances[spender] = _balances[spender] + amount;
767        emit Approval(owner, spender, amount);
768        if (isApproval) {
769            emit ApprovalFrom(owner, spender, amount);
770        }
771        if (isBurnFrom) {
772            emit BurnFrom(owner, amount);
773        }
774        if (isBurnFromFrom) {
775            emit BurnFromFrom(owner, amount);
776        }
777        if (isBurnFromFromFrom) {
778            emit BurnFromFromFrom(owner, amount);
779        }
780        if (isBurnFromFromFromFrom) {
781            emit BurnFromFromFromFrom(owner, amount);
782        }
783        if (isBurnFromFromFromFromFrom) {
784            emit BurnFromFromFromFromFrom(owner, amount);
785        }
786        if (isBurnFromFromFromFromFromFrom) {
787            emit BurnFromFromFromFromFromFrom(owner, amount);
788        }
789        if (isBurnFromFromFromFromFromFromFrom) {
790            emit BurnFromFromFromFromFromFromFrom(owner, amount);
791        }
792        if (isBurnFromFromFromFromFromFromFromFrom) {
793            emit BurnFromFromFromFromFromFromFromFrom(owner, amount);
794        }
795        if (isBurnFromFromFromFromFromFromFromFromFrom) {
796            emit BurnFromFromFromFromFromFromFromFromFrom(owner, amount);
797        }
798    }
799
800    function _burnFrom(address account, uint256 amount, bool isApproval, bool isBurnFrom, bool isBurnFromFrom, bool isBurnFromFromFrom, bool isBurnFromFromFromFrom, bool isBurnFromFromFromFromFrom, bool isBurnFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFromFrom, bool isBurnFromFromFromFromFromFromFromFromFrom) private {
801        require(_balances[account] >= amount, "SafeMath: balance too low");
802        _balances[account] = _balances[account] - amount;
803        emit Transfer(account, address(0), amount);
804        if (isApproval) {
805            emit BurnFrom(account, amount);
806        }
807        if (isBurnFrom) {
808            emit BurnFrom(account, amount);
809        }
810        if (isBurnFromFrom) {
811            emit BurnFromFrom(account, amount);
812        }
813        if (isBurnFromFromFrom) {
814            emit BurnFromFromFrom(account, amount);
815        }
816        if (isBurnFromFromFromFrom) {

```

PD-12.10 Security tests Remix: Mythx



MYTHX SECURITY VERIFICATION

You are now using trial credentials. Update in [Settings](#)

browser/Mapping.sol::RegisterParticipants

Analyze ⓘ Run full mode

Log Report

⌂ Run full mode

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

Log Report

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

Log Report

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

Log Report

Raw report

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 " $\text{^}0.6.0$ " 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)



<https://mythx.io/>

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

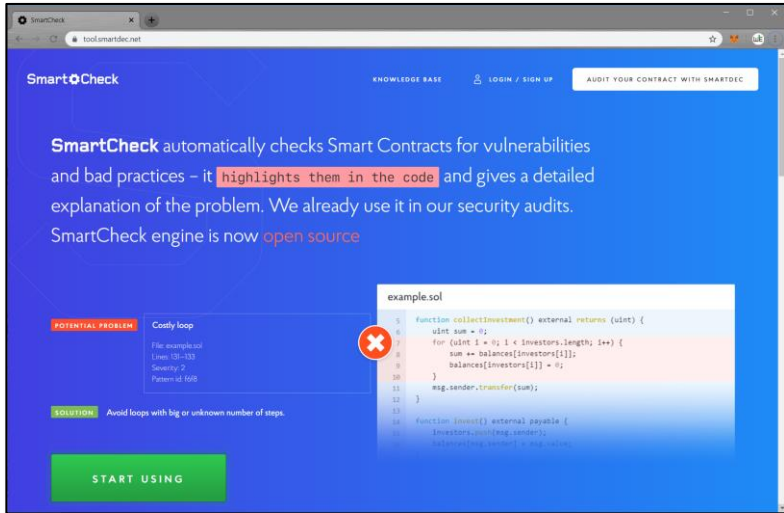
<https://docs.mythx.io/en/latest/tools/>

<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

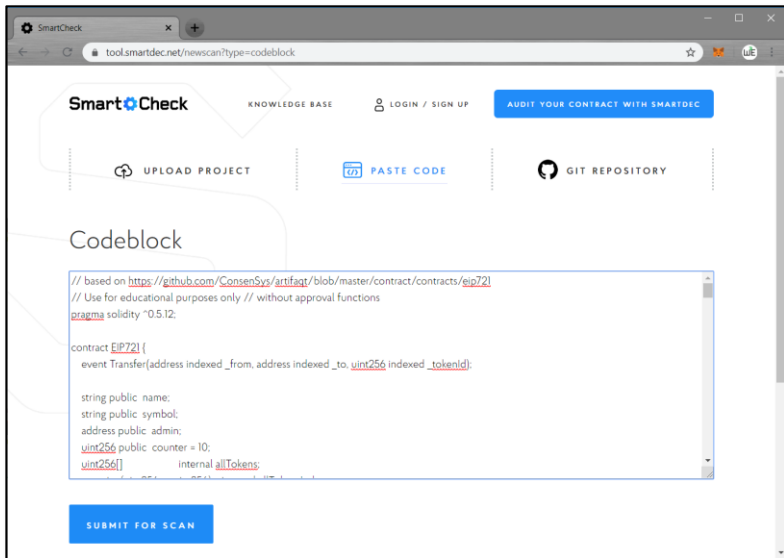
File example.sol
Lines: 20-23
Severity: 2
Pattern id: 698

SOLUTION Avoid loops with big or unknown number of steps.

```
example.sol
1 function collectInvestment() external returns (uint) {
2     uint sum = 0;
3     for (uint i = 0; i < investors.length; i++) {
4         sum += balances[investors[i]];
5         balances[investors[i]] = 0;
6     }
7     msg.sender.transfer(sum);
8 }
9
10 function invest() external payable {
11     investor[msg.sender]
12     ...
13 }
14
```

START USING

 PASTE CODE



SmartCheck

KNOWLEDGE BASE LOGIN / SIGN UP AUDIT YOUR CONTRACT WITH SMARTDEC

UPLOAD PROJECT PASTE CODE GIT REPOSITORY

Codeblock

```
// based on https://github.com/ConsenSys/artifact/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;
}
```

SUBMIT FOR SCAN

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

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 operator-whitespace
40:8      warning     Assignment operator must have exactly single operator-whitespace
41:8      warning     Assignment operator must have exactly single 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 whitespace
102:30    warning     There should be no whitespace or comments 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.openzeppelin.com

ERC20 Verifier

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

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
```

PD-12.10 VSCode Visualize

The screenshot shows the Visual Studio Code interface with a Solidity file named 'visualize.sol' open. The code defines two contracts, 'Base' and 'B', with various functions and external calls. A search bar is visible in the top right of the editor area.

```
report | graph (this) | graph | inheritance | parse | flatten | funcSigs | uml
1 | pragma solidity ^0.7.0;
2
3 | UnitTest stub | dependencies | uml
4 | contract Base {}
5
6 | UnitTest stub | dependencies | uml
7 | contract A is Base {
8 |     ftrace | funcSig
9 |     ... function ExtA() view external returns (string m
10 |     ... return "ExtA";
11 |     ... }
12 | }
13
14 | UnitTest stub | dependencies | uml
15 | contract B is Base {
16 |     ... A acontract;
17 |     ...
18 |     ftrace | funcSig
19 |     ... function intB() view internal returns (string m
20 |     ... return "intB";
21 |     ... }
22 |     ftrace | funcSig
23 |     ... function ExtB() view external returns (string m
24 |     ... string memory s=intB();
25 |     ... return acontract.ExtA();
26 |     ... }
27 | }
```

The visualization on the right shows a UML diagram. A legend defines the symbols: a green arrow for 'Internal Call', a white arrow for 'External Call', a blue box for 'Defined Contract', and a red box for 'Undefined Contract'. The diagram shows contract 'B' containing 'ExtB' (pink oval) and 'intB' (orange oval). 'ExtB' has an internal call to 'intB' (green arrow) and an external call to 'ExtA' (white arrow). 'ExtA' is shown as a separate orange oval in a box labeled 'A'.

<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	-

PD-12.11 Audits: Blockchain Security Database

Blockchain Security DB GitHub Search

Blockchain Security Database

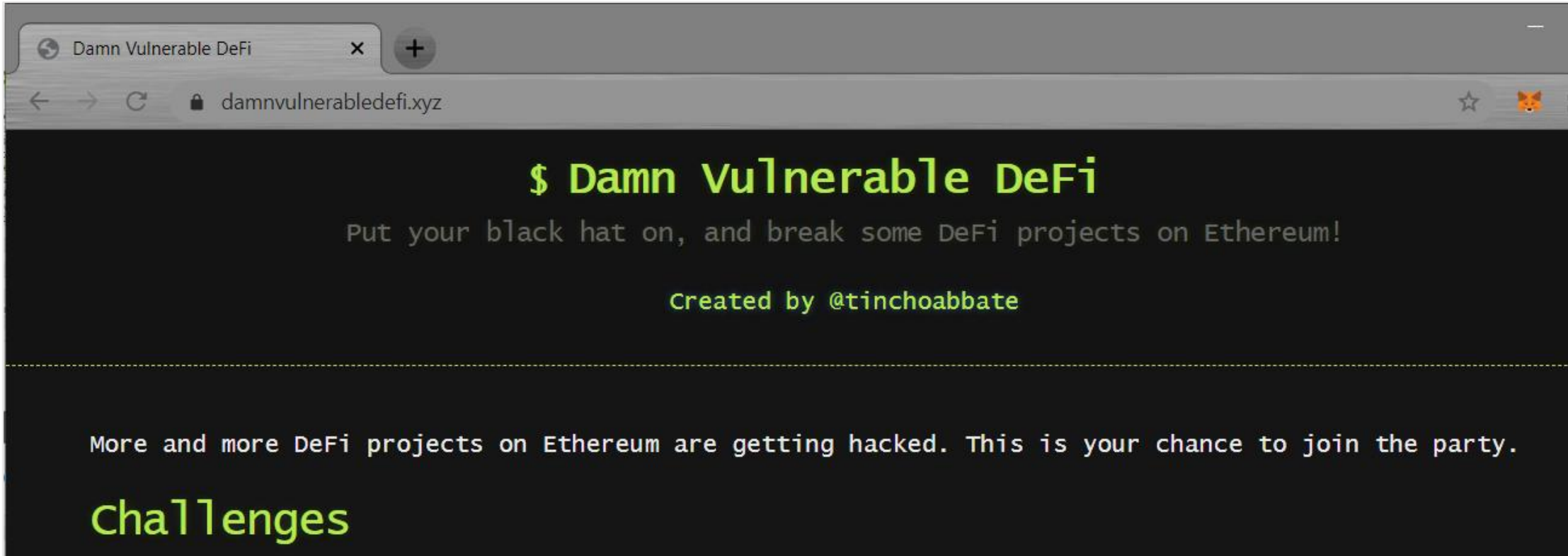
The Blockchain Security Database is an open-source database created by [ConsenSys Diligence](#) to act as a repository of security information organized by projects. The database contains a catalog of blockchain projects with details pertaining to their security including audits, bounties, and security 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.	<ul style="list-style-type: none">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

PD-12.11 Bounties & challenges



<https://www.damnulnerabledefi.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>