|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | **测试用例** | **测试指标** | **测试方法** |
| 7.1.0 区块链未知漏洞检测 | | | |
| 0-1 | privacy.go | 检测privacy.go智能合约，结果中应当包含由课题组首先发现并定义的chaincode-privacy-leakage-in-arg漏洞  （golang语言智能合约应当通过getTransit API传递隐私参数，如果不使用会造成参数中包含隐私数据，导致隐私泄露） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/blob/main/%E6%9C%AA%E7%9F%A5%E6%BC%8F%E6%B4%9E/privacy.go  2. 选择privacy.go将其代码复制到<http://39.103.152.161/>的合约代码中  3. 选择Golang，漏洞检测  4. 点击Analyze Now  5. 检测结果中包含chaincode-privacy-leakage-in-arg漏洞 |
| 0-2 | privacy.go | 检测privacy.go智能合约，结果中应当包含由课题组首先发现并定义的chaincode-privacy-leakage-in-ret漏洞  （golang语言智能合约的隐私查询函数应当用于不涉及区块链状态更改的函数，readWriteKVs函数中包含了PutState对区块链的状态进行更改，导致返回值中包含隐私数据，隐私泄露） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/blob/main/%E6%9C%AA%E7%9F%A5%E6%BC%8F%E6%B4%9E/privacy.go  2. 选择privacy.go将其代码复制到<http://39.103.152.161/>的合约代码中  3. 选择Golang，漏洞检测  4. 点击Analyze Now  5. 检测结果中包含chaincode-privacy-leakage-in-ret漏洞 |
| 0-3 | cbridge.sol | 检测cbridge.sol以太坊区块链程序，其中应该包含由课题组首先发现并定义的Cross-Chain Message Injection漏洞  （以太坊区块链跨链功能需要检查负责转账的智能合约的参数地址信息，否则会被攻击者利用发生注入攻击） | 1. 打开以太坊脆弱性发现工具V1.0  2. 选择cbridge.sol代码，点击运行按钮对其进行检测（该工具基于pythonV3.8.10在编译器PyCharm中开发）  3. 检测结果中包含Cross-Chain Message Injection漏洞 |
| 0-4 | Harmony.sol | 检测Harmony.sol以太坊区块链程序，其中应该包含由课题组首先发现并定义的Incomplete event漏洞  （以太坊区块链跨链功能需要原链产生状态更新后，提供本次更新的完整信息，否则目标链无法确认原链交易的具体情况，导致不一致） | 1. 打开以太坊脆弱性发现工具V1.0  2. 选择cbridge.sol代码，点击运行按钮对其进行检测（该工具基于pythonV3.8.10在编译器PyCharm中开发）  3. 检测结果中包含Incomplete event漏洞 |