|  |  |  |  |
| --- | --- | --- | --- |
| **用例编号** | **测试用例** | **测试指标** | **测试方法** |
| 7.1.1 区块链合约层漏洞检测（solidity语言智能合约） | | | |
| 1-1 | mycontract.sol | 检测mycontract.sol智能合约，结果中应当包含tx-origin漏洞  （Solidity语言智能合约将tx-origin用于require等用于权限控制语句，tx-origin的值会被攻击者操控从而绕过权限控制） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择mycontract.sol将其代码复制到[**http://39.103.152.161/**](http://39.103.152.161/)的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-2 | mycontract.sol | 检测mycontract.sol智能合约，结果中应当包含arbitrary-send漏洞  （Solidity语言智能合约中的转账函数sendTo使用任意的receiver作为参数，receiver的值会被攻击者操纵，从而将钱转入攻击者指定的账户） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择mycontract.sol将其代码复制到[**http://39.103.152.161/**](http://39.103.152.161/)的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-3 | token-with-backdoor.sol | 检测token-with-backdoor.sol智能合约，结果中应当包含backdoor漏洞（Solidity语言智能合约中存在公开的可被操控的关键函数） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择token-with-backdoor.sol将其代码复制到[**http://39.103.152.161/**](http://39.103.152.161/)的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-4 | crypto\_roulette.sol | 检测crypto\_roulette.sol智能合约，结果中应当包含uninitialized-storage漏洞（Solidity语言智能合约中存在未初始化的存储变量，对该变量的修改会覆盖其他变量的内容） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择crypto\_roulette.sol将其代码复制到[**http://39.103.152.161/**](http://39.103.152.161/)的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-5 | crypto\_roulette.sol | 检测crypto\_roulette.sol智能合约，结果中应当包含weak-prng漏洞  （Solidity语言智能合约中使用了不安全的随机因子block.number，该随机因子可以被攻击者操控） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择crypto\_roulette.sol将其代码复制到http://39.103.152.161/的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-6 | crypto\_roulette.sol | 检测crypto\_roulette.sol智能合约，结果中应当包含incorrect-equality漏洞  （Solidity语言智能合约中使用了严格等式作为判断条件，该判断条件容易被攻击者绕过） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择crypto\_roulette.sol将其代码复制到http://39.103.152.161/的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-7 | crypto\_roulette.sol | 检测crypto\_roulette.sol智能合约，结果中应当包含controlled-array-length漏洞  （Solidity语言智能合约中使用了可控制长度的数组，限制长度意外的部分容易被攻击者操控） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择crypto\_roulette.sol将其代码复制到http://39.103.152.161/的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-8 | rubixi.sol | 检测rubixi.sol智能合约，结果中应当包含unchecked-send漏洞  （Solidity语言智能合约中未检查转账函数send的返回值，如果交易失败则会导致金额丢失） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择rubixi.sol将其代码复制到http://39.103.152.161/的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-9 | rubixi.sol | 检测rubixi.sol智能合约，结果中应当包含divide-before-multiply漏洞  （Solidity语言智能合约中存在对一个整数先除后乘的操作，可能导致精度损失） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择rubixi.sol将其代码复制到http://39.103.152.161/的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-10 | proxy.sol | 检测proxy.sol智能合约，结果中应当包含controlled-delegatecall漏洞  （solidity智能合约中使用了可控制目标地址to的delegatecall，目标地址容易被攻击者操纵） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择proxy.sol将其代码复制到http://39.103.152.161/的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-11 | send\_loop.sol | 检测send\_loop.sol智能合约，结果中应当包含uninitialized-local漏洞  （Solidity语言智能合约中直接使用了未初始化的本地变量，执行结果可能指向未知目标） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择send\_loop.sol将其代码复制到http://39.103.152.161/的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-12 | send\_loop.sol | 检测send\_loop.sol智能合约，结果中应当包含uninitialized-state漏洞  （Solidity语言智能合约中存在未初始化的状态变量，该变量会按默认值参与运算，导致结果出错） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择send\_loop.sol将其代码复制到http://39.103.152.161/的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-13 | simple\_dao.sol | 检测simple\_dao.sol智能合约，结果中应当包含reentrancy-eth漏洞  （Solidity语言智能合约中存在重入漏洞） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择simple\_dao.sol将其代码复制到http://39.103.152.161/的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-14 | unchecked\_return\_value.sol | 检测unchecked\_return\_value.sol智能合约，结果中应当包含unchecked-lowlevel漏洞  （Solidity语言智能合约中未检调用函数call的返回值，如果交易失败，金额将被锁定在合约中） | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择unchecked\_return\_value.sol将其代码复制到http://39.103.152.161/的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |
| 1-15 | simple\_suicide.sol | 检测simple\_suicide.sol智能合约，结果中应当包含suicidal漏洞  （Solidity语言智能合约中存在未受保护的自毁指令，容易被攻击者操控 | 1. 打开网页  https://github.com/my-code-cloud/EvaluationCases/tree/main/solidity%E5%90%88%E7%BA%A6%E6%BC%8F%E6%B4%9E  2. 选择simple\_suicide.sol将其代码复制到http://39.103.152.161/的合约代码中  3. 选择solidity，漏洞检测  4. 点击Analyze Now |