**本体形式化验证工具检测项功能列表**

# 静态检测规则表

如表1-1，为本体形式化验证工具所覆盖的静态检测的规则库。其中包括三类主要的漏洞类型，第一为语言本身特性规定的漏洞类型，第二为平台对语言所做的特殊规定的漏洞类型，第三为平台规则库中所规定的漏洞类型。详细可参照《本体形式化工具功能测试用例说明》。

表1-1 本体形式化验证工具静态检测规则表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 检测项title | 类型type | 描述description | 示例example |
| 1 | Nonexistent-Operator | Error | Use of the non-existent {} operator  使用python语法中不支持的操作符 | 使用Python不支持的++或-- |
| 2 | Unreachable | Warning | Unreachable code  存在不可达流程 | 例如 程序块中break和return之后的语句无法执行 |
| 3 | Duplicate-key | Warning | Duplicate key {} in dictionary  字典数据类型中存在重复的key | 详见测试用例Duplicate-key |
| 4 | Using-Constant-Test | Warning | Using a conditional statement with a constant value  使用常量测试条件 | 详见测试用例Using-Constant-Test |
| 5 | Unnecessary-Pass | Warning | Unnecessary pass statement  使用多余的pass语句 | 详见测试用例Unnecessary-Pass |
| 6 | Invalid-Slice-Index | Error | Slice index is not an int, None  使用无效的切片索引操作字符串 | 例如使用字符串，id,None等方法作为切片索引 |
| 7 | Reimported | Warning | Reimport {} (imported line {})  模块重复导入 | 详见测试用例Reimported |
| 8 | Multiple-Imports | Warning | Multiple imports on one line ({})  在一行代码中使用import语句多次导入（应该使用from xx import xx） | 详见测试用例Multiple-Imports |
| 9 | Lack-Of-Event | Error | Should emit NEP5 event relate to method {}  缺少event | 例如缺乏transfer，transferfrom，approve事件调用 |
| 10 | Lack-Of-Witness | Error | The "from" address should be verified by CheckWitness  called合约地址验证 | 例如transfer，transferfrom，approve是否使用checkwitness对合约地址与from地址进行验证 |
| 11 | Unsupported-Operator | Error | Unsupported operator  使用不支持的string，bytearra，列表的操作符 | 例如在string，列表中使用+，\*，+=等操作符 |
| 12 | Unsupport-String-Formatted | Error | Unsupported string Format: {}  使用不支持的字符串格式化操作 | 详见测试用例Unsupport-String-Formatted |
| 13 | Find-Negative-Index-In-String-Or-Bytearray | Error | Avoid negative index in string or bytearray  String或者Bytearray切片索引为负数的情况 | 例如String或者Bytearray的切片索引通过计算公式传递负数 |
| 14 | Comparison-with-itself | Error | Comparison with itself  对象与自己本身进行比较 | 例如if a is a: |
| 15 | Assignment-from-no-return | Error | Assigning to function call which doesn't return  接收了没有返回值的函数 | 例如：简单合约的转账功能，TransferOntOng函数没有返回值，无法判断转账是否成功。 |
| 16 | Invalid-Sequence-Index | Error | Sequence index is not an int, slice  使用非int的类型做数组索引 | 例如使用id或者none做数组索引 |
| 17 | Assignment-from-None | Error | Assigning to function call which only returns None  return必为None | 例如：TransferOntOng函数存在return none |
| 18 | Unhashable-Dict-Key | Error | Dict key is unhashable  dict的key是不可用的 | 详见测试用例Unhashable-Dict-Key |
| 19 | No-Name-In-Module | Error | No name {} in module {}  导入了不存在的包 | 例如导入了不存在的模块，或者导入模块中不存在的函数。 |
| 20 | Unused-Variable | Warning | Unused variable {}  存在未使用的变量 | 详见测试用例Unused-Variable |
| 21 | Unused-Argument | Warning | Unused argument {}  存在未使用的参数 | 详见测试用例Unused-Argument |
| 22 | Redefined-Outer-Name | Error | Redefining name {} from outer scope (line {})  从外部范围重新定义 | 详见测试用例Redefined-Outer-Name |
| 23 | Not-support-slice-access | Error | Not support slice access  使用切片操作列表 | 例如：截取5名玩家，对数组切片 |
| 24 | Random-number-attack | warning | Random number can be predicted  随机数可能被预测造成攻击 | 详见测试用例Random-number-attack |
| 25 | Check-invoke-result | Warning | Invoke function return without check  没有对invoke返回值做判断 | 详见测试用例Check-invoke-result |
| 26 | No-except-handler | Warning | ONT do not support except handler  使用了为支持的except关键字作为异常处理 | 详见测试用例no except handler |
| 27 | No-return | Error | Return function which dosen’t return  返回没有返回值的函数 | 详见测试用例no-return |
| 28 | Only-slice-access | Error | Only slice access  使用索引操作字符串 | 详见测试用例only-slice-access |
| 29 | Possibly-unused-variable | Warning | Unused variable  存在未使用的变量 | 详见测试用例Possibly-unused-variable |
| 30 | Redefined-builtin | Error | Redefining built-in {}  重新定义本地内置函数 | 详见测试用例Redefined-builtin |
| 31 | Wrong-import-position | Error | Import "{}" should be placed at the top of the contract  from \*\* import 未放在首行 | 详见测试用例Wrong-import-position |
| 32 | Unused-wildcard-import | Warning | Unused import {} from wildcard  import导入多个模块未被使用 | 详见测试用例Unused-wildcard-import |
| 33 | Unused-Import | Warning | Unused import{}  导入的模块未被使用 | 详见测试用例Unused-Import |
| 34 | Slice-elem-out-of-bounds | Error | Slice elem out of bounds  切片索引越界 | 例如字符串end索引为函数，函数返回越界 |

# 动态检测规则库

如表1-2为本体形式化检测工具所覆盖的动态检测规则表，主要为在动态执行过程中会出现的漏洞，能够覆盖静态检测无法覆盖的内容。与静态检测相辅相成相得益彰。详细可参照《本体形式化工具功能测试用例说明》。

表1-2 本体形式化验证工具动态检测规则表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 检测项title | 类型type | 描述description | 示例example |
| 1 | Div-zero-occurred | Error | Div zero  存在零除错误 | 详见测试用例Div-zero-occurred |
| 2 | Index-out-of-range | Error | Index out of range  存在数组越界 | 详见测试用例index out of range |
| 3 | Assert-fail | Error | Assert conditions are not always satisfied 功能安全属性约束不满足 | 例如：assert（a>10）但是在程序运行过程中不能始终满足这个断言条件。 |
| 4 | Require-fail | Error | Require Conditions are not satisfied  条件表达式的成立条件永远无法满足 | 例如：require（a>10）但是在程序运行过程中该条件始终不满足。 |
| 5 | Integer-overflow-occurred | Error | Integer may overflow  变量运算中可能存在整形上溢的错误 | 例如：int\_8 a = 126;  int\_8 b = 10;  a = a+b;  若没有使用safemath保证安全，这里会出现整型上溢漏洞 |
| 6 | Integer-underflow-occurred | Error | Integer may underflow  变量运算中可能存在整型下溢的错误 | 例如  int\_8 a = -127;  Int\_8 b = 10;  a = a - b; 若没有使用safemath保证安全，这里会出现整型下溢漏洞 |
| 7 | Data-injection-attack | Error | Data injection attack | 详见测试用例Data injection attack |