<https://blog.soliditylang.org/2023/11/08/verbatim-invalid-deduplication-bug/>

要纯yul代码才能触发，无

<https://soliditylang.org/blog/2023/07/19/full-inliner-non-expression-split-argument-evaluation-order-bug/>

自定义优化器序列会触发，无

<https://soliditylang.org/blog/2023/07/19/missing-side-effects-on-selector-access-bug/>

selector相关，无

<https://soliditylang.org/blog/2022/09/08/storage-write-removal-before-conditional-termination/>

assembly中加入 return语句

<https://soliditylang.org/blog/2022/08/08/calldata-tuple-reencoding-head-overflow-bug/>

函数的最后一个参数是uint或bytes32的calldata数组，需要用到结构体，无

如：

Struct T{ …} Function f(bool a,T calldat b,bytes32[2] calldata c) public {}

<https://soliditylang.org/blog/2022/06/15/dirty-bytes-array-to-storage-bug/>

bytes s; s.push(); 空的push会导致将脏数据复制到storage类型的变量中

<https://soliditylang.org/blog/2022/06/15/inline-assembly-memory-side-effects-bug/>

在assembly中加入mstore和mload

<https://soliditylang.org/blog/2022/05/17/data-location-inheritance-bug/>

需要合约的继承，无

<https://soliditylang.org/blog/2022/05/17/calldata-reencode-size-check-bug/>

被external函数调用嵌套数组，无

<https://soliditylang.org/blog/2022/03/16/encodecall-bug/>

使用abi.encodeCall(f,(…)),其中函数f有一个参数为类型bytesNN。无

<https://soliditylang.org/blog/2021/09/29/user-defined-value-types-bug/>

存储布局的改变，之前小于32字节的类型的变量总是会占用整个存储槽，现在，多个小于32字节的类型的变量可以共享一个存储槽。无

<https://soliditylang.org/blog/2021/09/29/signed-immutables-bug/>

contract C {

int8 immutable x = -2;

function f() public view returns (bytes32 r) {

int8 y = x;

assembly { r := y }

}

}

Immutable的storage变量，在存储转换时会发生错误。

<https://soliditylang.org/blog/2021/04/21/decoding-from-memory-bug/>

**The bug is present in all prior versions of ABI coder v2. 无**

<https://soliditylang.org/blog/2021/03/23/keccak-optimizer-bug/>

在assembly中加入a:=keccake256(0,32);

<https://soliditylang.org/blog/2020/10/19/empty-byte-array-copy-bug/>

* You copy an **empty** bytes or string value from memory or calldata to storage.
* You extend the storage value by modifying .length or using .push() (not .push(c)).
* You read the newly created byte array element without writing to it first.

<https://soliditylang.org/blog/2020/04/06/memory-creation-overflow-bug/>

定义一个使用用户提供长度的数组。无

<https://soliditylang.org/blog/2019/06/25/solidity-storage-array-bugs/>

If you have deployed contracts which use signed integer arrays in storage and either directly assign

* a literal array with at least one negative value in it (x = [-1, -2, -3];) or
* an existing array of a *different* signed integer type

to it, this will lead to data corruption in the storage array.

<https://soliditylang.org/blog/2019/03/26/solidity-optimizer-and-abiencoderv2-bug/>

需要外部调用函数。无

<https://soliditylang.org/blog/2018/09/13/solidity-bugfix-release/>

结构体作为event的参数会出问题。无

<https://soliditylang.org/blog/2017/05/03/solidity-optimizer-bug/>

一个常量以0xfff..开始，以一长串零为结尾；

这个常数在多个位置使用，这样在使用优化器优化时会出现问题。

<https://soliditylang.org/blog/2016/11/01/security-alert-solidity-variables-can-overwritten-storage/>

如果大于类型运行的值赋值给第一个变量，则该值会覆盖第二个值