Post

Chaofan Shou @shouccccc

Follow

The following code in Vyper compiler caused $24M+ hacked in multiple @CurveFinance pools

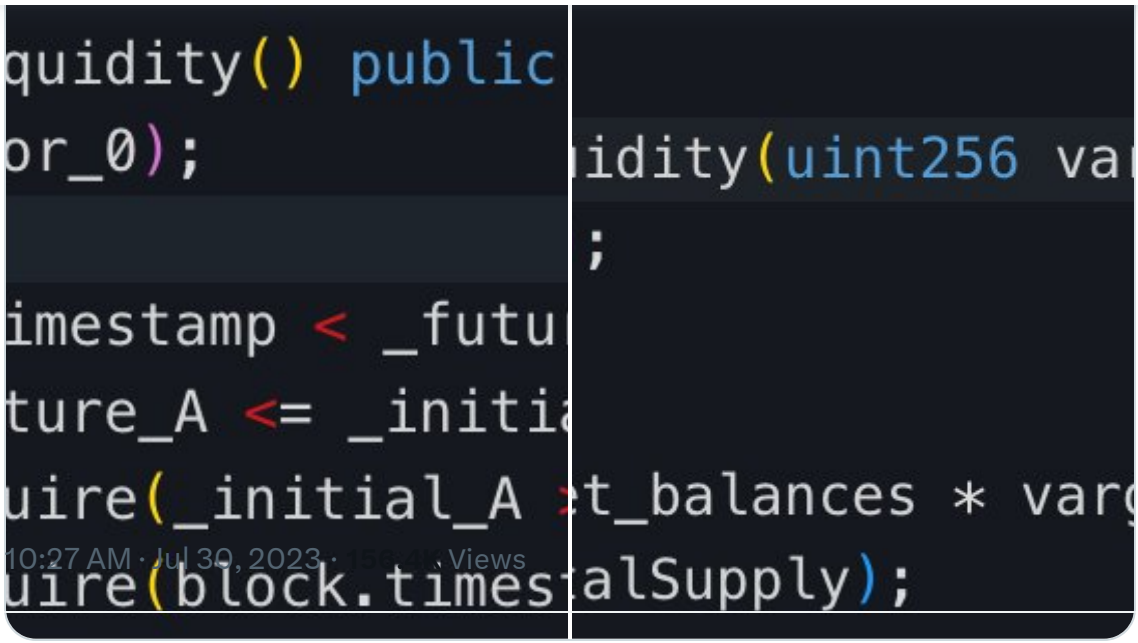
It seems the nonreentrant guard uses different storage slot.

```
def set_storage_slots(vyper_module: vy_ast.Module) -> None:
    """
    Parse module-level Vyper AST to calculate the layout of storage variab
    """
    # Allocate storage slots from 0
    # note storage is word-addressable, not byte-addressable
    storage_slot = 0

    for node in vyper_module.get_children(vy_ast.FunctionDef):
        type_ = node._metadata["type"]
        if type_.nonreentrant is not None:
            type_.set_reentrancy_key_position(StorageSlot(storage_slot))
            # TODO use one byte - or bit - per reentrancy key
            # requires either an extra SLOAD or caching the value of the
            # location in memory at entrance
            storage_slot += 1

    for node in vyper_module.get_children(vy_ast.AnnAssign):
        type_ = node.target._metadata["type"]
        type_.set_position(StorageSlot(storage_slot))
        # CMC 2021-07-23 note that HashMaps get assigned a slot here.
        # I'm not sure if it's safe to avoid allocating that slot
        # for HashMaps because downstream code might use the slot
        # ID as a salt.
        storage_slot += math.ceil(type_.size_in_bytes / 32)
```

> **Tony KΞ** @tonyke_bot · Jul 30, 2023
>
> Certain type of Curve factory pool is encountering read-only reentrancy attack and causing a total loss of $11m(@JPEGd_69) + $13m(@AlchemixFi) + ...
>
> Initial investigation founds that vyper compiler (0.2.15) doesn't implement the reentrancy guard correctly. ...
>
> Show more



81 Reposts **32** Quotes **236** Likes **73** Bookmarks

73