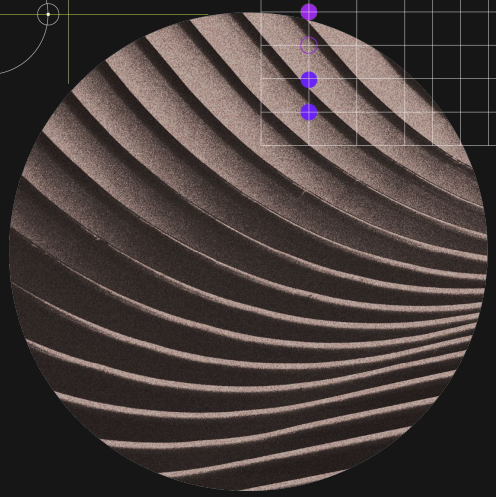
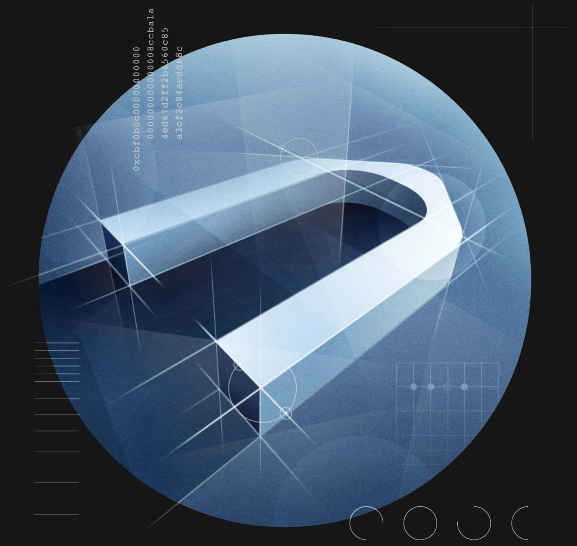


0xcbf0b0c0000000000
0000000000000008ccba1a
4ed67d2f2b6560c85
a3cf2c84acdd68c

Thanks for all the bugs!

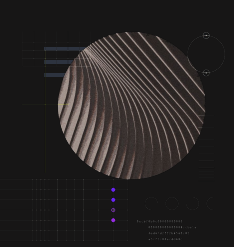
Yannis Smaragdakis

DEDAUB



DEDAUB

contract-library.com



Our Research and Technology

- Static Analysis
  - trying to create a model of all possible program behaviors
- All analyses specified declaratively
  - logical rules (thousands of them)

```
LoopBoundBy(loop, var) ←
  InductionVar(i, loop),
  !InductionVar(var, loop),
  Flows(var, condVar),
  Flows(i, condVar),
  LoopExitCond(condVar, loop).
```

Vulnerability Disclosures

Vulnerability Disclosures (since 2021)

Yield Skimming: Forcing Bad Swaps on Yield Farming

Vulnerability Disclosures (since 2021)

Yield Skimming: Forcing
Swaps on Yield Farming

Inside the War Room That Saved Primitive Finance

Jonah Michaels Follow

Mar 4 · 12 min read

Vulnerability Disclosures (since 2021)

Yield Skimming: Forcing Swaps on Yield Farming

Inside the War Primitive Fina

Jonah Michaels Follow

Mar 4 · 12 min read

"Look ma', no source!" Hacking a DeFi Service with No Source Code Available

Vulnerability Disclosures (since 2021)

Ethereum Pawn Stars: “$5.7M in hard assets? Best I can do is $2.3M”

e the War
itive Fina
ichaels Follow
2 min read

“Look ma’, no source!” Hacking a DeFi Service with No Source Code Available

Vulnerability Disclosures (since 2021)

Ethereum Pawn Stars: “$5
hard assets? Best I can do i
$2.3M”

Killing a Bad (Arbitrage) Bot

... to save its owners

urce!” Hacking a
n No Source

Vulnerability Disclosures (since 2021)

Ethereum Pawn Stars: “$5
hard assets? Best I can do i
$2.3M”

Vulnerability Disclosures (since 2021)

Phantom Functions and the Billion-Dollar No-op

By the Dedaub team

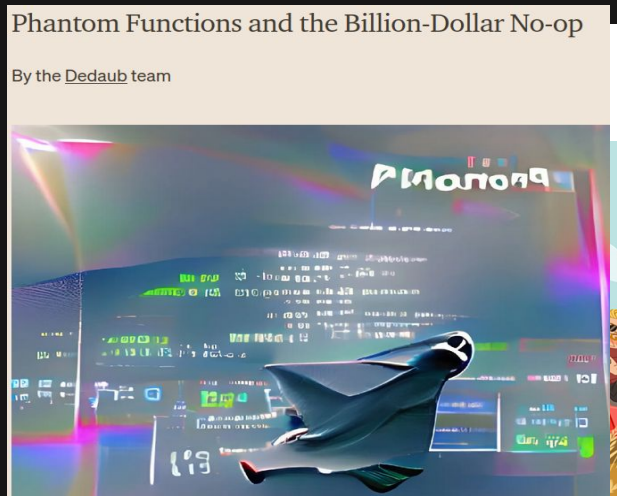
Harvest Finance

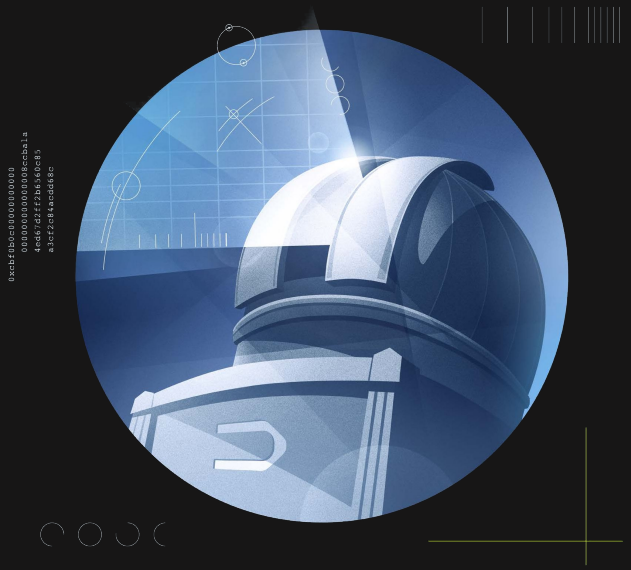
Uninitialized Proxies

Bug Fix Postmortem

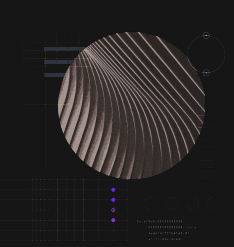
Vulnerability Disclosures (since 2021)

- Several major security vulnerabilities, 9 bug bounties of ~$3M total
  - by DeFi Saver, Dinngo/Furucombo, Primitive, Armor, Vesper, BT Finance, Harvest, Multichain/Anyswap, Rari/Tribe DAO



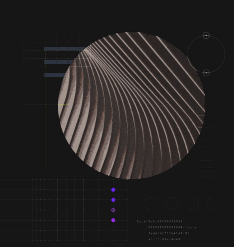


Solidity/EVM Traps



Solidity/EVM Traps: `this`

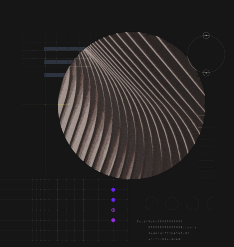
- In most OO languages
  `this.fun();`
  and
  `fun();`
  are synonyms



Solidity/EVM Traps: `this`

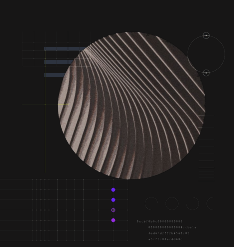
- In most OO languages
  `this.fun();`
  and
  `fun();`
  are synonyms

- Not in Solidity: external call, `msg.sender` changes



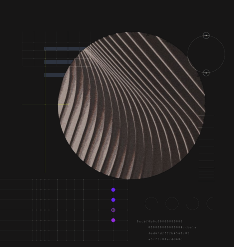
Solidity/EVM Traps: Phantom Functions

- The source of our largest vulnerability to date
  - >$1B value



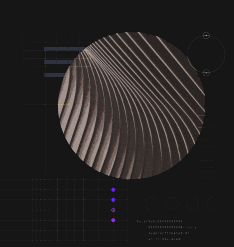
The Billion-Dollar Bug

```
function deposit() external { …
  underlying.transferFrom(msg.sender, this, amount);
  …
}

function depositWithPermit(…) external {
  underlying.permit(…, v, r, s);
  underlying.transferFrom(target, this, amount);
}
```



Phantom Permit (in WETH9)

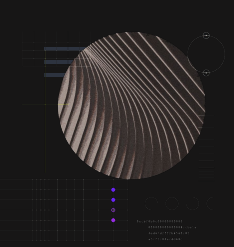
```
function() public payable {
  deposit();
}
```



Phantom Permit (in WETH9)

```
function() public payable {
  deposit();  // not reverting
}
```

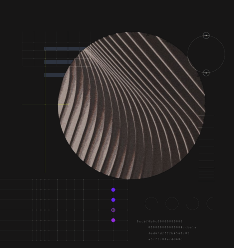
A phantom function is one that the contract does not define but accepts calls to it without reverting



The Billion-Dollar Bug

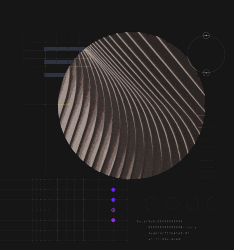
```
function deposit() external { …
  underlying.transferFrom(msg.sender, this, amount);
  …
}

function depositWithPermit(…) external {
  underlying.permit(…, v, r, s);
  underlying.transferFrom(target, this, amount);
}
```

Writing
~~Correct~~
~~Secure~~
Well-Auditable
Code



Unpleasant Truth #1

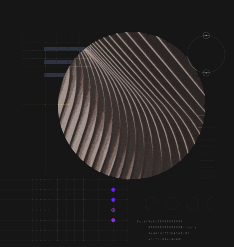
- *Not all audits are equal:*
  there is a wide range in the auditors' confidence
  - an audit stops when the time allotted is over, not when confidence is 100%
    - *confidence is never 100%*



Lots of Good Practices for High Audit Confidence

- Documentation
- Comments!
- Informative variable names
- Well-tested code
- Good communication channel

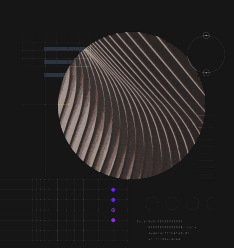
Most top-quality projects do ok in all these



Big Difference-Maker: Invariants

Properties that should hold no matter what

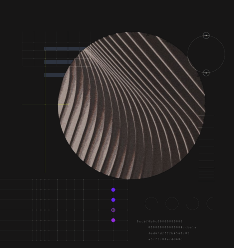
- Temporal/State:
  - “*we only get here with* `lock` *called before*”
  - “*only reachable from methods called by the hub*”
  - “*no ETH balance to remain here*”



Big Difference-Maker: Invariants

Properties that should hold no matter what

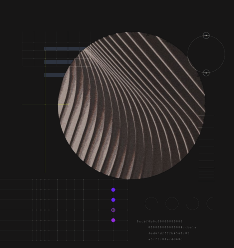
- Functional
  - “*denominated in the last of the tokens in the array*”
  - “*if positive it means input amt, if negative, output*”
  - “*e5 scale*”



Big Difference-Maker: Invariants

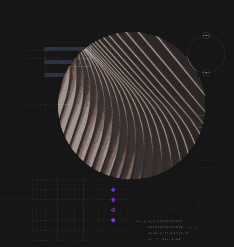
Properties that should hold no matter what

- Structural
  - “*strategies have to override functions* `reinvest` *and* `reportProfit`”
  - “*if a strategy issues rewards, it should override the* `rewardToken` *function AND the* `rewardRate` *function*”



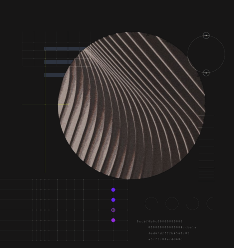
Example Documentation/Invariants

- `subUntilZero(x,y)`
  // returns pair of two elements:
  // (how much is left of x if we subtract y,
  // how much are we missing from x to reach y).
  // One of the two is zero
- `t.liquidityLowerD8`
  // A delta of how much liquidity is added when crossing
  // this tick left to right. Meaning, we go from tick `t-1`, to `t`,
  // we add this much to `currentLiquidity`



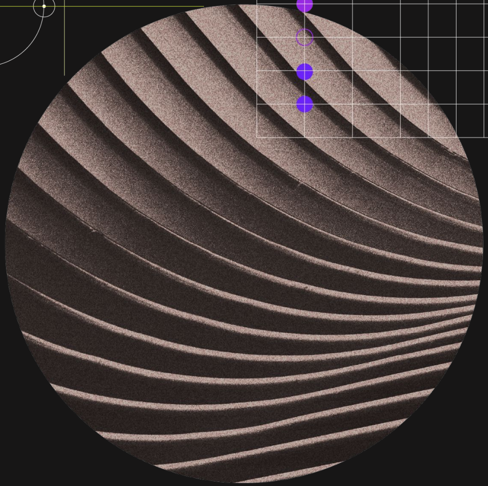
Invariants

- `liquidityOutside`
  // “outside” = the region from the tick to infinity (positive
  // or negative) *away* from the current tick.
  // E.g., if the current tick is below tick t, then “outside” is
  // from t to positive infinity. If it’s above,
  // “outside” is from negative infinity to t.
- `curRewardAmount`
  // rewards per token accumulated since the beginning
  // of time, both paid out and not



Unpleasant Truth #2

- *Auditors are not owners of your project, you are!*
  - the auditor can help you see things differently
  - but you have to follow the questions deeply yourself, not do the least possible just to dismiss them



0xcbf0b0c000000000
00000000000008ccba1a
4ed67d2ff2b6560c85
a3cf2c84acdd68c

Thanks for all the bugs!

Yannis Smaragdakis

DEDAUB

contract-library.com