Why Forks Fail Often

Changing Risks with Unchanged Code

World’s Fastest Intro

I’m engn33r and I do security

yAudit

yAcademy

Slides QR Code

https://engn33r.com/forksfail.pdf

Security

When can secure code become insecure?

1. When secure contract code is changed (duh!)
2. When something OUTSIDE the secure contract code is changed

Context Makes Security Hard

Blockchain tech moves fast

Any change can cause issues

Security scope is **NOT** only your smart contract code

Butterfly effect: a change in one protocol can impact another elsewhere

DeFi money legos? Or DeFi quicksand?

How Crypto Security Audits Work

Crypto Twitter Audits Shiller

Audits Aren’t Bulletproof

Audit scope = code written by the protocol developers

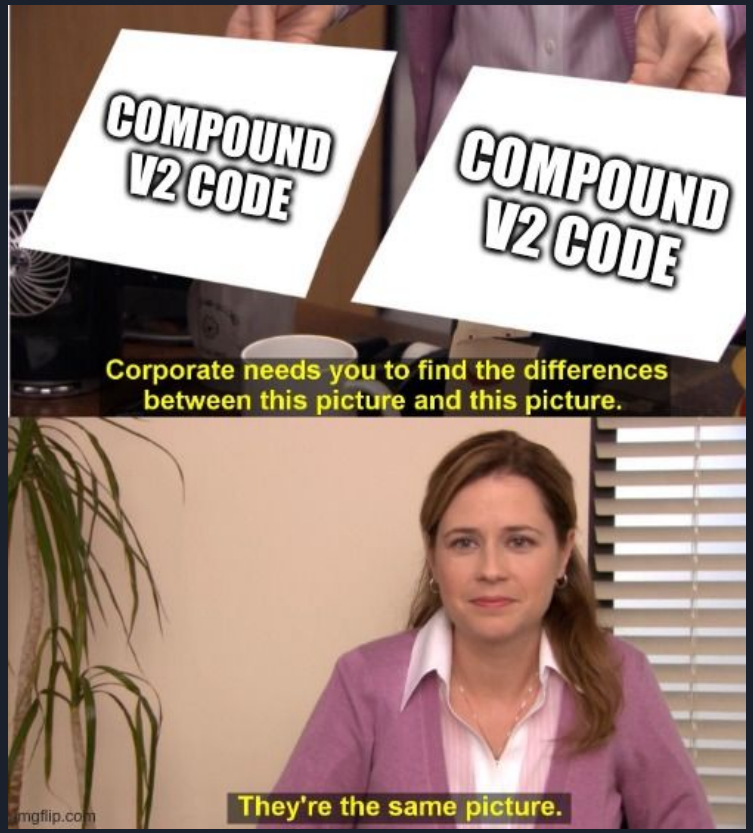
“Outside of scope” often ignored, can still cause big problems

Misperception of “if my code is bug-free, it’s safe”

Forking Secure Code

Raise your hand if you think forking secure code is always safe and secure!

Forking Secure Code



Forking Secure Code

- CREAM: flashloan attack & reentrancy with ERC777-like token (no checks-effects-interaction protection) Postmortem POC
- CREAM: Price manipulation Postmortem POC
- Lendf.me: Flashloan and reentrancy (no checks-effects-interaction protection) Postmortem
- Compound: Double-entry point token issue Retrospective POC
- Lodestar Finance: Exchange rate manipulation Thread POC
- Hundred Finance: Flashloan and reentrancy on gnosis, where native token has callback hook (no checks-effects-interaction protection) Postmortem
- Ola Finance: Flashloan and reentrancy (no checks-effects-interaction protection) Postmortem
- Rari Capital: Flashloan and reentrancy (no checks-effects-interaction protection) POC
- Venus: Chainlink LUNA oracle became inaccurate during the Terra collapse, which cause a similar result as oracle manipulation and led to draining of protocols writeup
- Hundred Finance: Exploit of empty markets Postmortem POC
- 0VIX: price oracle vulnerability allowed donation-based price maniulation Thread POC
- Midas Capital: Exploit of empty markets writeup
- Onyx Finance: Exploit of empty markets Postmortem POC
- Sonne Finance: Exploit of empty markets Postmortem

Screenshot from https://github.com/YAcademy-Residents/defi-fork-bugs

Context is EVERYTHING

When is secure code not secure? When the context changes!

Changes that add risk:

1. Governance design changes
2. Governance decision changes
3. Blockchain parameter changes
4. EVM chain differences and changes
5. Other protocol changes

1. Governance Design Changes

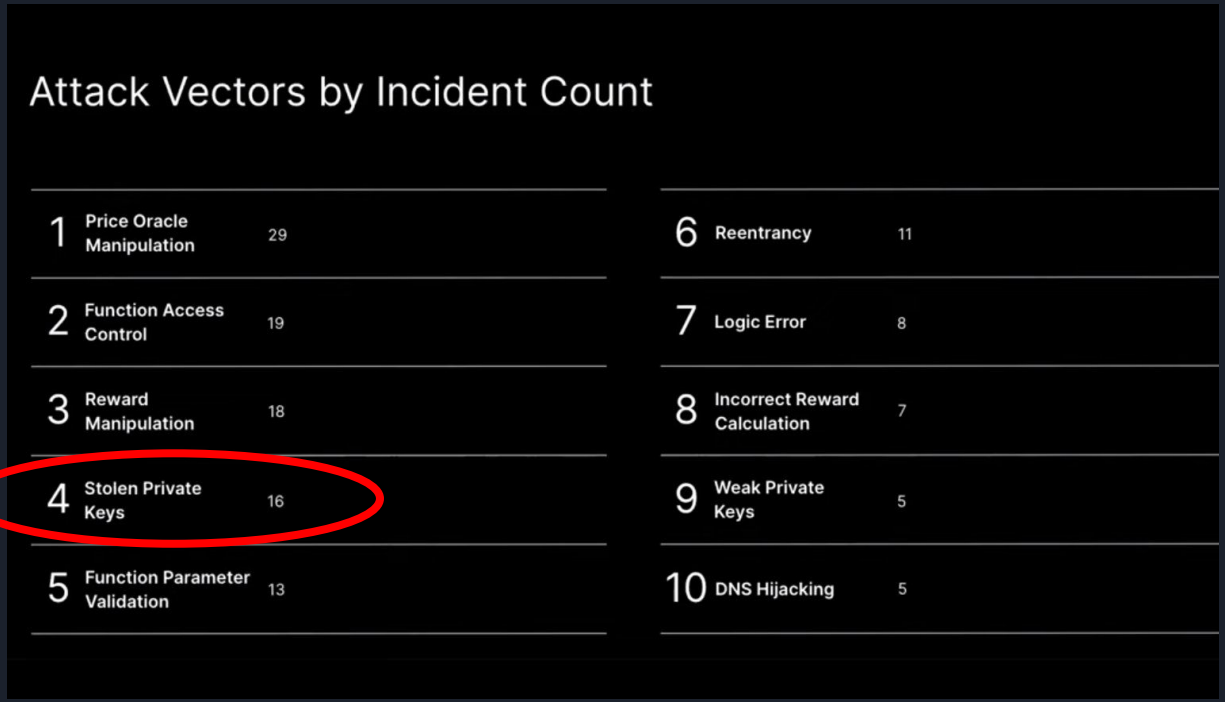
EOA address (AKA not a multisig) != multisig

1 of 1 multisig != 5 of 8 multisig

8 multisig addresses all owned by 1 person != 8 person decentralized multisig

Deploying w/proxy contracts != deploying w/o proxies

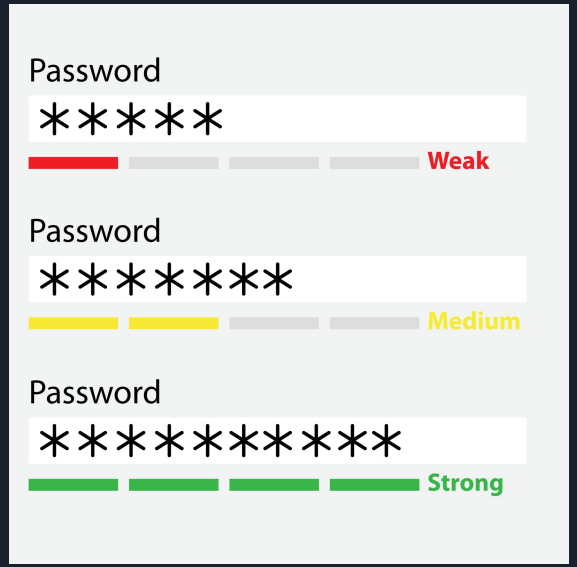
1. Governance Design Changes



Data from BlockThreat by @_iphelix https://substack.com/@blockthreat

1. Governance Design Changes

Web2 equivalent of weak governance is weak passwords



2. Governance Decision Changes

2. Governance Decision Changes

Example #1:

- Governance adds support for a new token in the protocol

What can go wrong:

- Supporting a reentrant token (big problem for Compound v2 forks)
- Supporting a token with an insecure price data source (very bad)
- Supporting weird ERC20 tokens that the design cannot support (e.g. rebasing tokens, fee on transfer) may break protocol accounting

  https://github.com/d-xo/weird-erc20

2. Governance Decision Changes

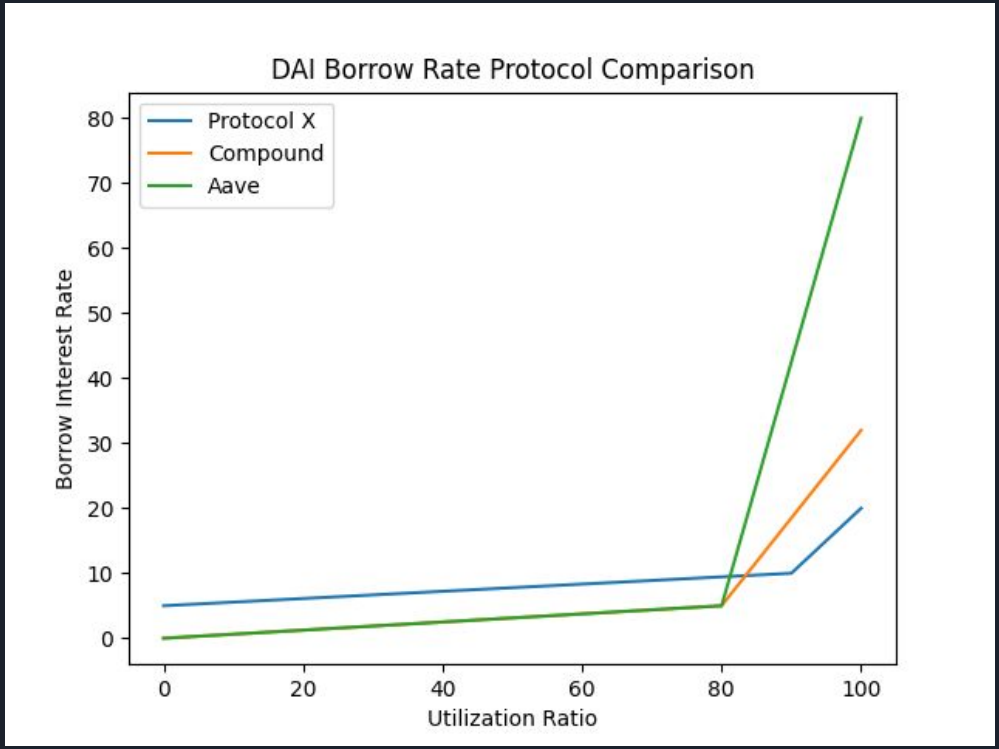
Example #2:

- Lending protocols need governance to update interest rate models based on market conditions

What can go wrong:

- Interest rate model increases the risk of bad debt in the protocol
- Interest rate model reduces the interest that depositors receive

2. Governance Decision Changes



2. Governance Decision Changes

The only (?!) example of security guidance on this topic:

https://hackernoon.com/how-to-review-a-governance-action

3. Blockchain Parameter Changes

Example: Ethereum's switch from PoW to PoS ("The Merge") impacted the security assumptions of TWAP oracle manipulation: https://blog.uniswap.org/uniswap-v3-oracles

Block	Age	Txn	Fee Recipient
19904531	3 mins ago	123	beaverbuild
19904530	3 mins ago	134	beaverbuild
19904529	3 mins ago	137	beaverbuild
19904528	3 mins ago	177	beaverbuild
19904527	4 mins ago	141	beaverbuild

3. Blockchain Parameter Changes

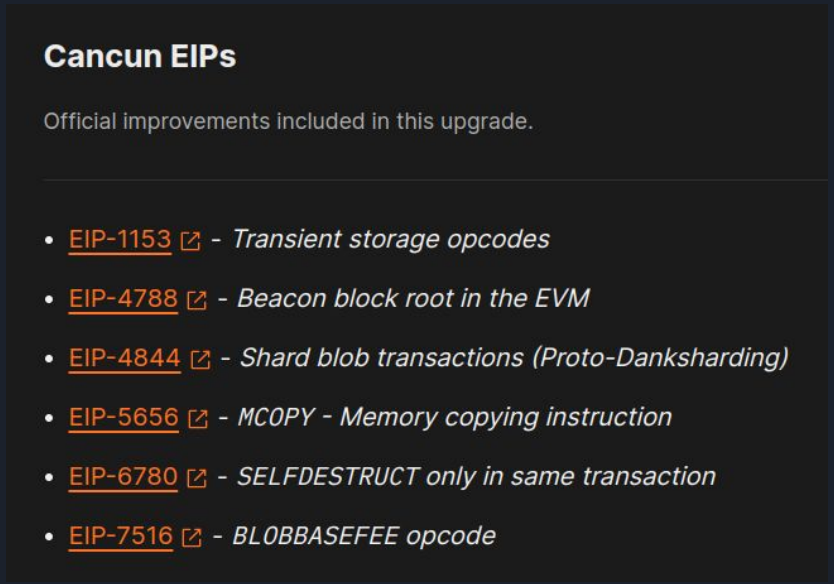
Example: EIP-1153 added new TSTORE and TLOAD opcodes

This can introduce a new reentrancy risk in some cases

https://chainsecurity.com/tstore-low-gas-reentrancy/

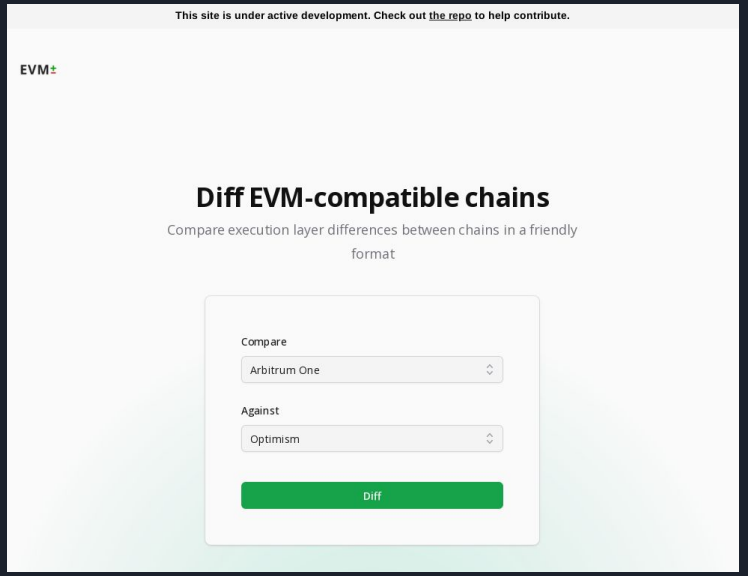
3. Blockchain Parameter Changes

Most blockchains are still changing, as new EIPs indicate



4. EVM Chain Differences

“EVM compatible” chains are increasingly fractured https://www.evmdiff.com/



4. EVM Chain Differences

4. EVM Chain Differences

EIP implementation is not synchronized between chains

- EIP-1559 (gas fee pricing change)
  - Mainnet Ethereum: August 5 2021
  - Polygon: January 18 2022
  - Optimism: June 6 2023
  - BNB: August 30 2023

4. EVM Chain Differences

EIP-1559 Parameters

The base fee on OP Mainnet is, like Ethereum, computed via the EIP-1559 ↗ mechanism. The EIP-1559 parameters used by OP Mainnet differ from those used by Ethereum as follows.

Parameter	OP Mainnet value	Ethereum value (for reference)
Block gas limit	30,000,000 gas	30,000,000 gas
Block gas target	5,000,000 gas	15,000,000 gas
EIP-1559 elasticity multiplier	6	2
EIP-1559 denominator	250	8
Maximum base fee increase (per block)	2%	12.5%
Maximum base fee decrease (per block)	0.4%	12.5%
Block time in seconds	2	12

From https://docs.optimism.io/chain/differences

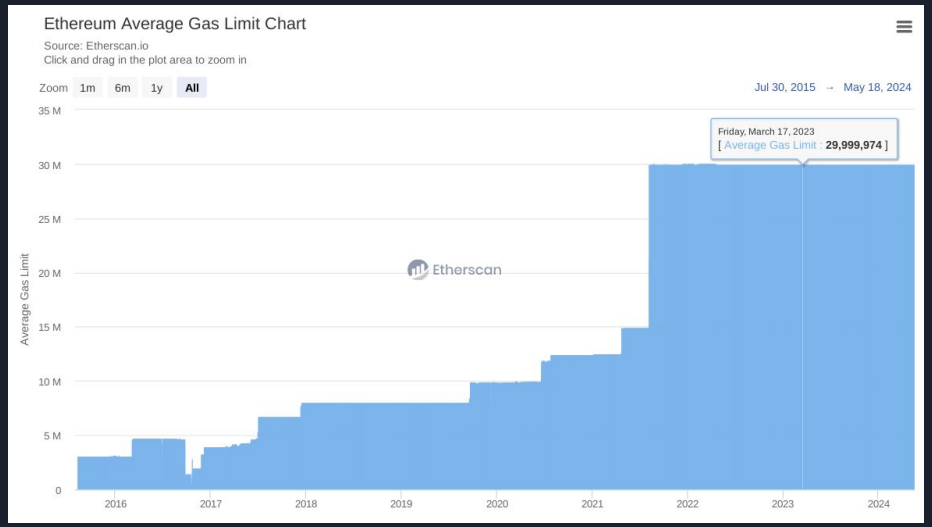
4. EVM Chain Differences

Other recent EIPs with possible impact:

- EIP-3855: Add PUSH0 opcode
- EIP-1153: Add TSTORE and TLOAD opcode
- EIP-3541: prevent contracts starting with 0xEF

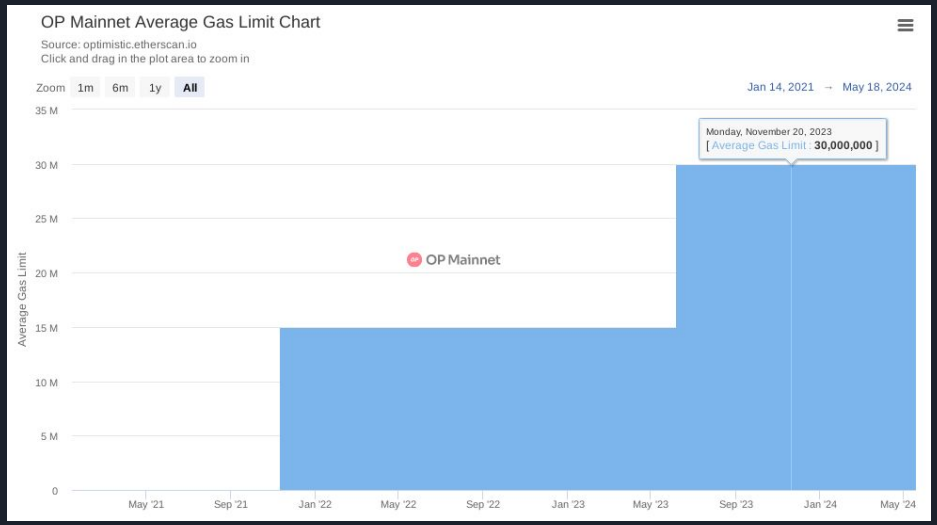
4. EVM Chain Differences

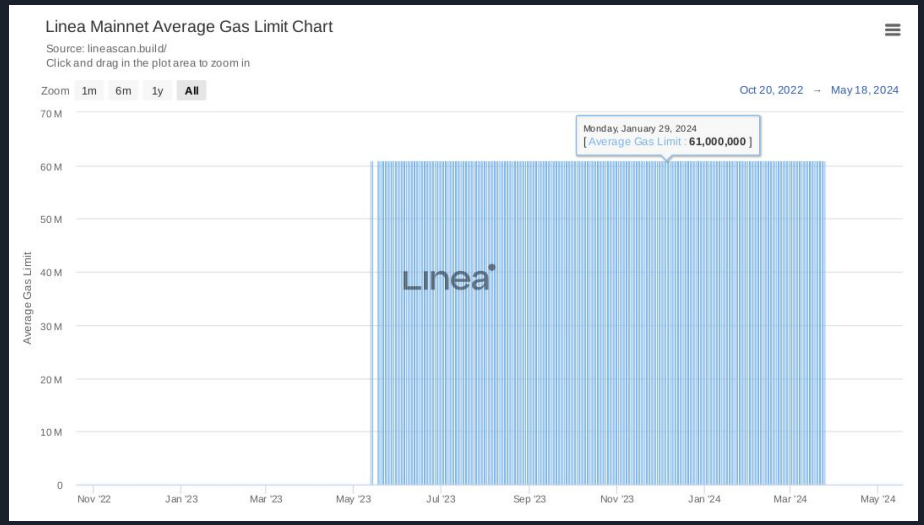
Gas limit differences



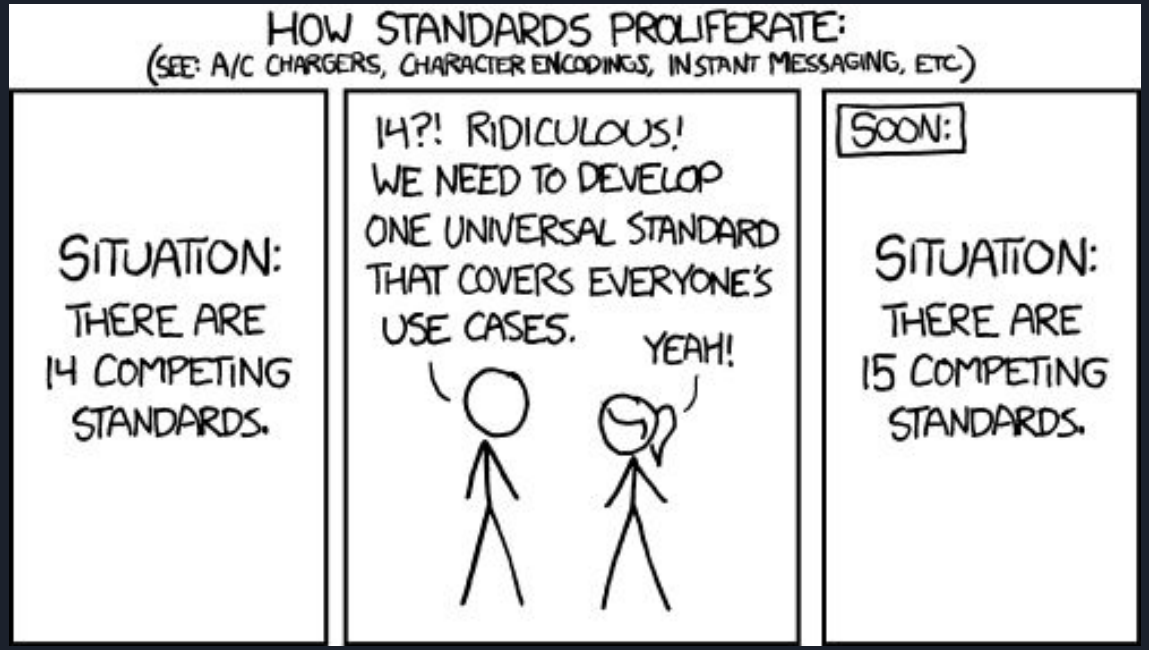
4. EVM Chain Differences

Gas limit differences





4. EVM Chain Differences



5. Other Protocol Changes

DeFi money legos is the ideal, DeFi quicksand may be the reality

Example: Aave vs. Morpho debate led to new Merit rewards alignment

"*Morpho optimizers are a leech on top of the Aave protocol*"

5. Other Protocol Changes

Introduction of flashloans increased the risks to other protocols

Cream Finance Exploited in Flash Loan Attack Netting Over $100M

Binance Smart Chain hackers made $167M with flash loans, exploits in May

Euler Finance hacked for over $195M in a flash loan attack

DeFi protocol Platypus suffers $8.5M flash loan attack, suspect identified

Value DeFi Suffers $6M Flash Loan Attack

$182 million stolen from stablecoin provider Beanstalk Farms in 'flash loan' attack

5. Other Protocol Changes

The discovery of new types of hacks can make similar protocols vulnerable

Mostly a change in ecosystem knowledge

Example: A fork getting hacked can cause copycat hacks of other forks

After the “Compound fork empty markets” bug was first exploited, other Compound forks that made the same mistake were quickly exploited

Lots of risks, what to do

Can all these risks be prevented before the protocol is deployed?

No!

So should we give up?

No!

What's the answer?

Security extends beyond an audit!

Suggestions

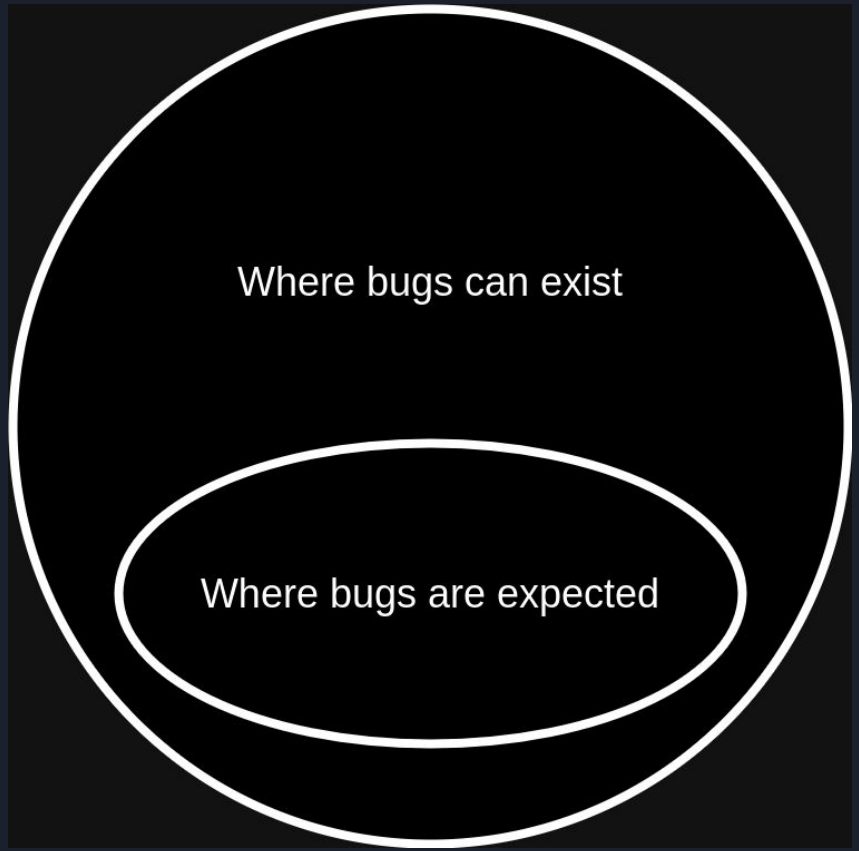
What smart contract developers can do:

- Include the deployment script in the security audit scope
- Make plans early for which chains the contracts will be deployed
- Create a governance decision review and testing process

After launch, watch for:

- New EIPs that may have impact
- Changes in any interconnected protocols (DeFi legos)
- New hacks that may apply to your protocol

Learnings



Summary

Changes OUTSIDE contract code can add risk:

1. Governance design changes
2. Governance decision changes
3. Blockchain parameter changes
4. EVM chain differences and changes
5. Other protocol changes

Remember: **Context is EVERYTHING**