CANTINA

Web3 Bug Bounties

Nov 28th, 2025

CANTINA

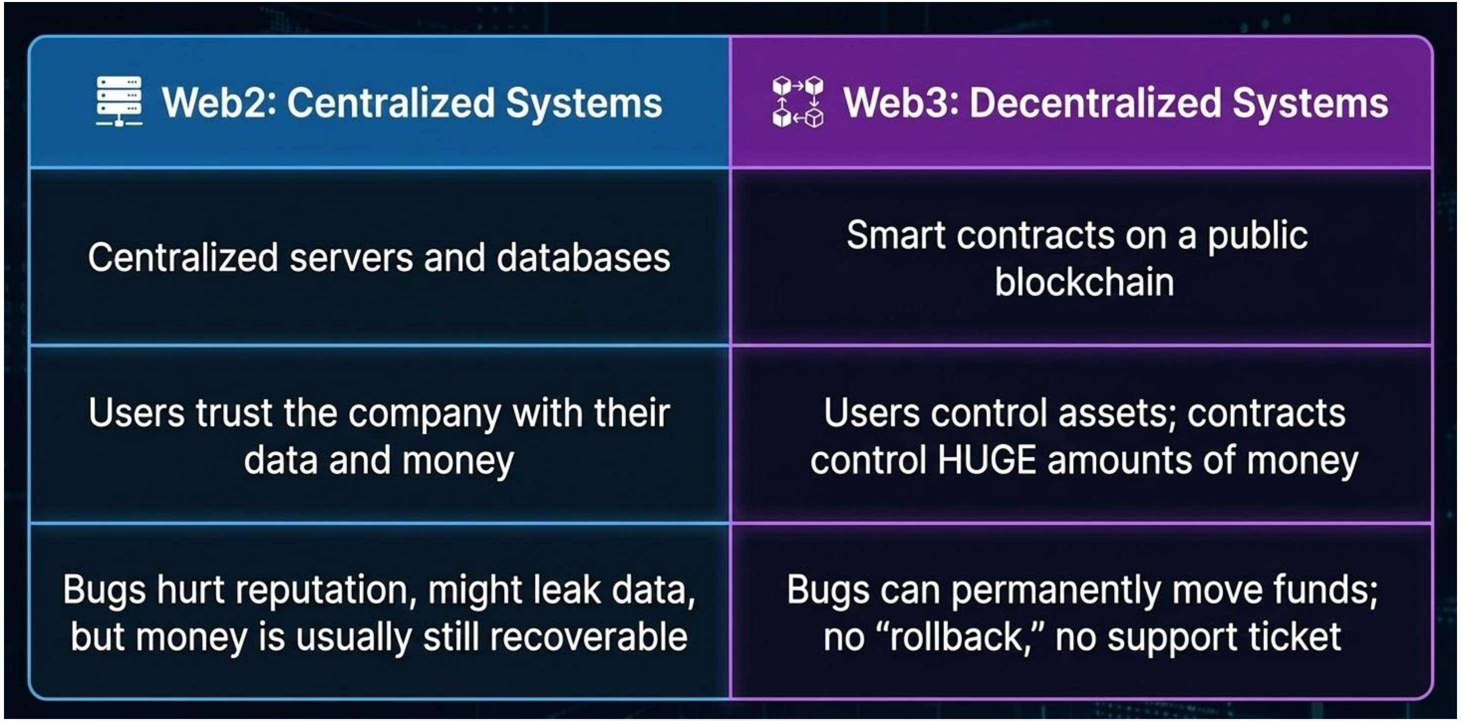
whoami

- Core team for Spearbit/Cantina
- Joined Web3 security in early 2022
- Spearbit has secured +2B in TVL over different engagements in the past 4 years
- X/Twitter: 0xMorph

Prerequisites Knowledge

- What is Web3?
- How does bug bounties work?
- How does severities work in web3?

Main differences between web2 and web3



Web2: Centralized Systems	Web3: Decentralized Systems
Centralized servers and databases	Smart contracts on a public blockchain
Users trust the company with their data and money	Users control assets; contracts control HUGE amounts of money
Bugs hurt reputation, might leak data, but money is usually still recoverable	Bugs can permanently move funds; no “rollback,” no support ticket

Why BBPs in Web3 pay so much compared to Web2

Open-source by default ⇒ everyone sees the code

DeFi/NFT/Games often manage millions in TVL

Projects are under time pressure, ship fast, and make mistakes

Bounties are cheaper than getting hacked

Public optics: “We paid a bounty” > “We got drained and blamed the hacker”

Public optics:
“We paid a bounty” > “We got drained and blamed the hacker”

Web3 Attack Surfaces compared to web2

Smart contracts
(Solidity / EVM bytecode)

Off-chain infrastructure:

- APIs, backend services, bots / keepers
- Frontend / websites of dApps
- Wallet integrations & signing flows
- Governance (voting, proposals, multisigs)

Classic Web2 Bugs that Still Matter in Web3

- **XSS** / **CSRF** on dApp frontends (steal signatures, change destination addresses)
- Misconfigured APIs or admin panels that control contracts / oracles
- Access control on dashboards that trigger on-chain actions
- **Bad auth** around anything that can sign or send transactions

- Access control on dashboards that trigger on-chain actions
- **Bad auth** around anything that can sign or send transactions

Web3 Specific Bug classes

Reentrancy (call back into contract mid-execution)

Broken access control / privilege checks in contracts

Price oracle manipulation & flash-loan abuse

Integer issues (overflows, underflows, precision)

Logic errors in financial math (fees, interest, reward distribution)

Signature misuse / replay / wrong domain separation

Hands-on Bug Bounty Methodology

Where bounties live:

- Platforms (Immunefi, Cantina, Hackenproof, etc.)
- Direct programs / security pages

Typical flow:

Find suspicious behavior or code pattern

↓

Reproduce with a minimal PoC (often on a fork of mainnet)

↓

Show realistic financial impact

↓

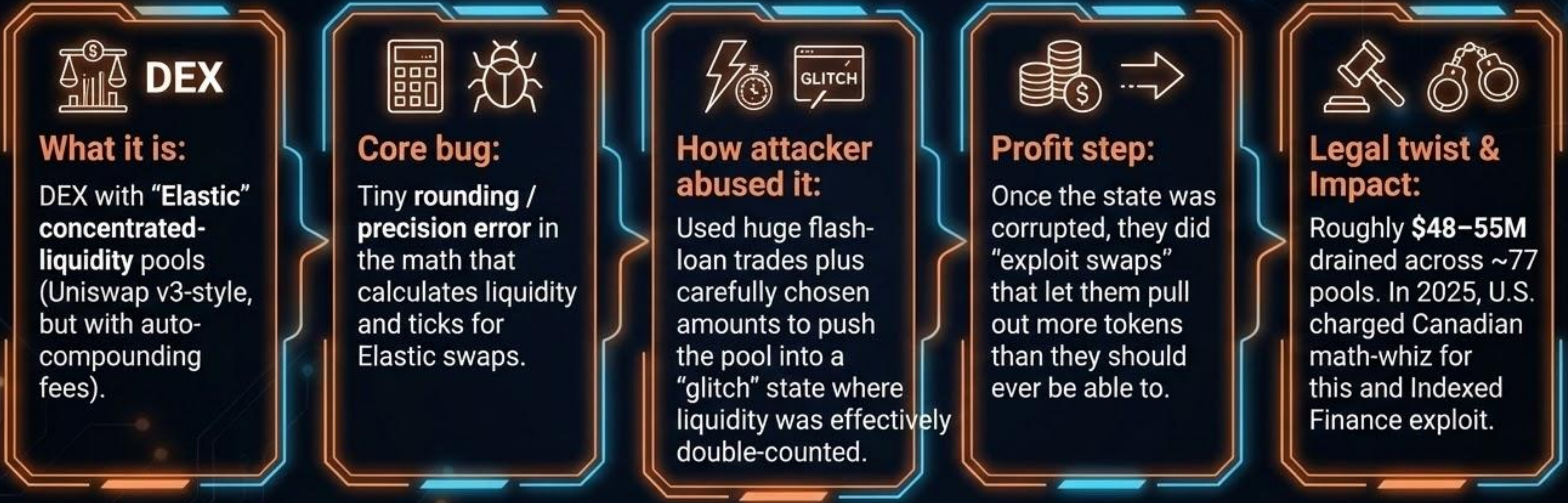
Privately disclose via platform / contact

Differences vs Web2:

- On-chain reproduction
- Emphasis on dollar impact
- Sometimes negotiations around "white-hat" rescue vs bounty

Case Study

KyberSwap – Elastic Exploit (Nov 2023)



What it is:

DEX with “**Elastic**” **concentrated-liquidity** pools (Uniswap v3-style, but with auto-compounding fees).

Core bug:

Tiny **rounding / precision error** in the math that calculates liquidity and ticks for Elastic swaps.

How attacker abused it:

Used huge flash-loan trades plus carefully chosen amounts to push the pool into a “glitch” state where liquidity was effectively double-counted.

Profit step:

Once the state was corrupted, they did “exploit swaps” that let them pull out more tokens than they should ever be able to.

Legal twist & Impact:

Roughly **$48–55M** drained across ~77 pools. In 2025, U.S. charged Canadian math-whiz for this and Indexed Finance exploit.

> Nice talking angle: **“One rounding bug, tens of millions gone.”**

CANTINA

Case Study

GMX V1: Vulnerability & Attack

What it is: Perp DEX; traders leverage, LPs hold GLP token backing trades.

Design quirk: Zero price impact – trades filled at oracle price, not on-chain order book.

Attack idea: Move AVAX price on CEXs, get big AVAX perp fills on GMX at old oracle price.

Execution Flow

1. Open large AVAX positions on GMX.
2. Push AVAX price up or down on CEXs.
3. Close positions on GMX at favorable oracle prices.
4. Repeat the loop a few times.

Outcome & Aftermath

Outcome: Trader walked away with **~$500k–700k**; losses eaten by GLP LPs.

Aftermath: GMX capped AVAX open interest & adjusted risk – design/economics exploit, not smart-contract bug.

> **“GMX worked exactly as designed – that was the problem.”**

Classic hack (2025): GMX V1 also had a **$42M GLP vault exploit** (cross-contract reentrancy + bad AUM math).

Q&A