What is a flash loan attack?

A flash loan attack uses a flash loan, an uncollateralized loan that must be borrowed and repaid within a single transaction, to give an attacker enormous temporary capital to manipulate a protocol. With millions in hand for one transaction, the attacker can skew a price oracle, imbalance a pool, or trigger faulty logic, extract profit, and repay the loan, all atomically. Flash loans are not the bug; they remove the cost of capital, exposing protocols that assumed attackers could not move large sums. The defense is robust, manipulation-resistant design.

HOW IT WORKS

01 How it works and example

A typical flash-loan-powered exploit, all in one transaction:

1. Borrow a large sum via flash loan. 2. Manipulate: dump the borrowed funds into a low-liquidity pool to crash or spike a price the target reads as an oracle. 3. Exploit: interact with the target protocol while the price is wrong, for example borrow far more than the collateral is really worth, or mint/redeem at a distorted rate. 4. Repay the flash loan and keep the profit.

Flash loans also amplify governance attacks (borrow tokens to pass a vote). Documented for defensive context.

> IT REMOVES THE COST OF CAPITAL
>
> *Flash loans are not themselves a vulnerability, they just mean an attacker has unlimited capital for one transaction. Any protocol that was "safe" only because manipulating it costs a lot of money is now exposed.*

HOW TO DEFEND

- Use manipulation-resistant price feeds: decentralized oracles and time-weighted average prices (TWAP), never a single spot price from a low-liquidity pool.
- Do not trust in-transaction spot prices for critical accounting.
- Add economic guardrails: caps, circuit breakers, and sanity checks on large swings.
- Make governance flash-loan-resistant (voting snapshots, timelocks) so borrowed tokens cannot pass votes.
- Audit the economics, model what an attacker with unlimited one-transaction capital can do.

SOURCES

[1] OWASP Smart Contract Top 10

[2] Ethereum.org: Smart contract security

[3] SWC Registry: Smart Contract Weakness Classification