Advanced Topics: Scaling, Exploits, and Smart Contract Security

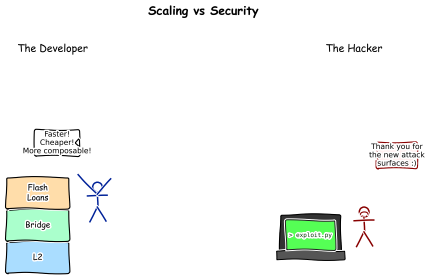
Layer 2 Solutions, Flash Loans, MEV, and the Art of Building Secure Protocols

Prof. Dr. Jörg Osterrieder

BSc Blockchain, Crypto Economy & NFTs

Spring 2026

[Cartoon] Faster, Cheaper, Hackable: Pick Two



Blockchain promised unstoppable money. Then we wanted it faster, so we built Layer 2 networks. Then we wanted more features, so we added bridges. Then we wanted zero-capital trading, so we invented flash loans.

Every new feature created a new attack surface.

The bridges we built to move assets between chains became the most-hacked infrastructure in crypto. The composability that lets protocols snap together like Legos also lets attackers chain exploits across three protocols in a single transaction.

The core tension of this lecture: How do you build systems that are both unstoppable and safe — when every new feature creates a new attack surface?

Since 2020, bridge exploits alone have caused over $2.5 billion in losses — more than all bank robberies in the US combined

Learning Objectives

By the end of this lecture, you will be able to:

1. Describe the scalability trilemma and explain how Layer 2 solutions address it *[Understand]*
2. Compare optimistic rollups and ZK-rollups on security, finality, and cost *[Analyze]*
3. Calculate fee savings from blob transactions after the Dencun upgrade *[Apply]*
4. Trace a flash loan attack step-by-step and identify the exploit mechanism *[Apply]*
5. Evaluate a smart contract's security posture using audit frameworks *[Evaluate]*

Prerequisites: Basic blockchain concepts (Module A) and DeFi fundamentals (Module E).

This lecture synthesizes L41 (Layer 2 Scaling), L42 (Flash Loans & MEV), and L43 (Smart Contract Security)

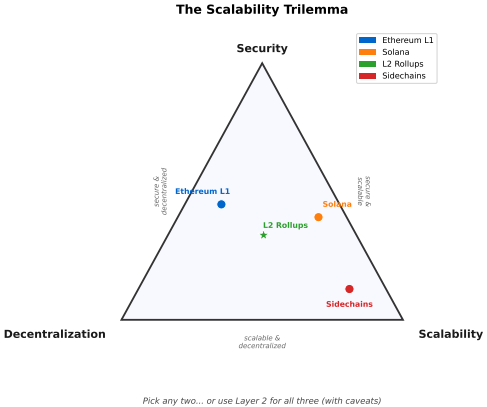
The Advanced Topics Landscape

These three topics form a connected chain:

1. **Layer 2 Scaling** creates new infrastructure (rollups, bridges) to make blockchain faster and cheaper
2. **Flash Loans & MEV** exploit the composability and atomicity of that infrastructure
3. **Smart Contract Security** develops defenses against those exploits

Each topic motivates the next: scaling creates attack surfaces, attacks demand security tools, and security constraints shape how we scale.

You cannot understand any one of these topics without the other two.



The scalability trilemma: every blockchain must sacrifice at least one of security, scalability, or decentralization

Your Smart Contract Just Got Hacked

Imagine you deployed a lending protocol last week. TVL hit $50M overnight. You celebrated. Your team was ecstatic. Then you get a Discord message at 3am:

"Someone drained the entire pool in a single transaction."

You pull up the block explorer. The attacker borrowed $500M with zero collateral from a flash loan provider, manipulated your price oracle by dumping tokens into a low-liquidity pool, borrowed against the inflated collateral in your protocol, and walked away with $50M in profit — all in 13 seconds, within a single atomic transaction.

Your code compiled. Your tests passed. Your audit was clean. But the attack vector was a **combination of three features you never tested together**: flash loans, oracle dependency, and cross-protocol composability.

The attacker used no private keys, no stolen credentials, no social engineering. They used your own protocol's logic against itself.

This lecture teaches you why this happens, how it works, and what you can do about it.

This is not hypothetical — Beanstalk ($182M, April 2022) and Cream Finance ($130M, Oct 2021) were attacked exactly this way

The Scalability Problem

Scalability Trilemma

A blockchain network can optimize for at most two of three properties simultaneously: **Security**, **Scalability**, and **Decentralization**. Improving one requires weakening at least one other.

Analogy: Like a triangle where stretching one corner pulls the others. You can make a blockchain faster, but only by either reducing the number of validators (less decentralized) or weakening the consensus mechanism (less secure).

Ethereum Layer 1 today:

- Throughput: approximately 15 transactions per second
- Gas fees: $5–$50 during congestion
- Block time: 12 seconds
- Full node storage: over 1 TB

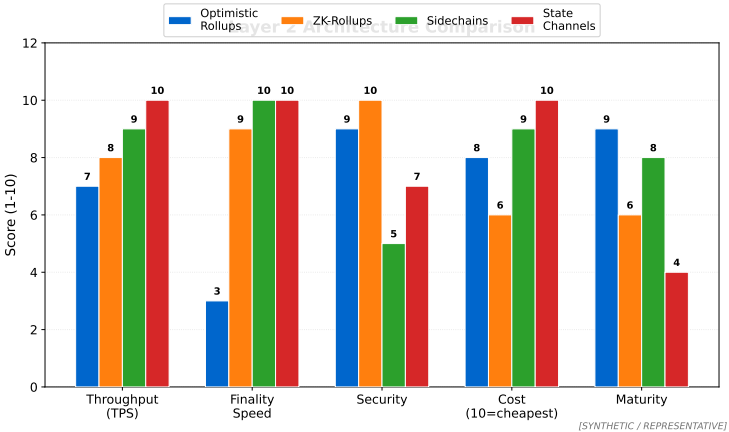
For comparison:

- Visa: 65,000 TPS capacity
- PayPal: 1,000+ TPS average

Layer 2 solutions try to break the trilemma by processing transactions off-chain while inheriting L1 security.

Ethereum's 15 TPS means a popular NFT mint can push gas fees above $100 — pricing out most users

Layer 2 Solutions: Three Approaches



[SYNTHETIC / REPRESENTATIVE]

Rollups:
Bundle hundreds of transactions off-chain, post compressed data to L1. Inherit Ethereum's full security.

Sidechains:
Independent chains with their own validators. Bridge assets to/from Ethereum. Own security model.

State Channels:
Two parties transact privately off-chain, settle the final state on L1. Best for repeated interactions.

Rollups dominate the L2 landscape because they inherit Ethereum's security guarantees — sidechains do not

Optimistic vs ZK-Rollups

Property	Optimistic Rollups	ZK-Rollups
How it works	Assume transactions valid; allow 7-day challenge period	Generate a mathematical proof that all transactions are valid
Security model	Fraud proofs (at least 1 honest verifier needed)	Validity proofs (math guarantees correctness, no trust needed)
Finality time	7 days (challenge window)	Minutes (once proof verified)
Current TPS	2,000–4,000	2,000–10,000
Example protocols	Arbitrum, Optimism, Base	Starknet, zkSync, Scroll
EVM compatible?	Yes (native)	Improving (zkEVM)
Main trade-off	Slow withdrawals to L1	Complex prover infrastructure

Key insight: Optimistic rollups are simpler to build but have a 7-day withdrawal delay. ZK-rollups are harder to build but provide instant mathematical guarantees of correctness.

The industry consensus (2025–2026): ZK is the long-term winner, but optimistic rollups have a head start in adoption and TVL.

Arbitrum holds approximately 45% of all L2 TVL as of early 2026 — but ZK-rollups are the fastest-growing segment

Flash Loans: Atomic Uncollateralized Borrowing

Flash Loan

A flash loan allows a user to borrow any amount of cryptocurrency with zero collateral, use it within a single transaction, and repay it before the transaction finalizes. If the borrower cannot repay, the entire transaction reverts as if nothing happened.

Key property: ATOMIC — all-or-nothing.

The loan exists only for approximately 13 seconds (one block). The blockchain either sees the full cycle (borrow, use, repay) or it sees nothing at all.

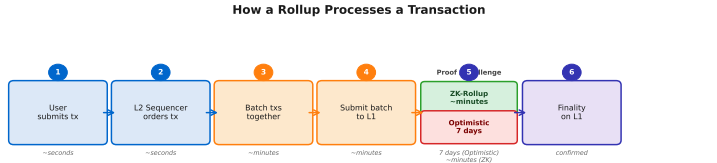
Legitimate uses:

- **Arbitrage:** Exploit price differences across exchanges in one transaction
- **Collateral swaps:** Change your collateral type without closing a position
- **Self-liquidation:** Repay your own loan to avoid liquidation penalties
- **Governance:** Temporarily acquire voting power (controversial)

The innovation: Flash loans make capital efficiency infinite — you can use $500M for 13 seconds and pay only a 0.09% fee.

Aave processes over $10B in flash loans per month — most are legitimate arbitrage, but the exploit potential is enormous

How a Rollup Transaction Works

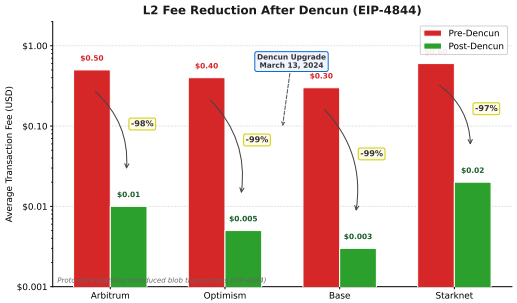


- **What you see:** A step-by-step flow from user submission to final settlement on Ethereum L1
- **Key pattern:** Execution happens off-chain (fast and cheap), but data and proofs are posted to L1 (secure and permanent)
- **Takeaway:** Rollups do not sacrifice L1 security — they inherit it by anchoring compressed transaction data to Ethereum

---

The sequencer collects and orders transactions off-chain, then posts batches to L1 — users pay L2 fees, not L1 gas

Worked Example: L2 Fee Savings After Dencun



The Dencun upgrade (March 13, 2024) introduced *blob transactions* (EIP-4844): a new, cheaper way for rollups to post data to Ethereum.

Worked calculation:

$$\underbrace{\text{Total Fee}}_{\text{what you pay}} = \underbrace{\text{L2 execution}}_{\text{computation}} + \underbrace{\text{L1 data posting}}_{\text{security anchor}}$$

Before Dencun:
100 swaps on Arbitrum at \$0.50 each = **\$50**
(L1 calldata costs dominated the fee)

After Dencun:
100 swaps on Arbitrum at \$0.01 each = **\$1**
(Blob data is 10–100x cheaper than calldata)

Savings: 98% reduction in L2 fees.

Blob transactions store data temporarily (approximately 18 days) instead of permanently, which is sufficient for rollup security.

EIP-4844 (proto-danksharding) was the single largest fee reduction in Ethereum's history — L2 fees dropped overnight

Worked Example: Flash Loan Arbitrage

Scenario: Token X trades at $10.00 on DEX A and $10.50 on DEX B. You have zero starting capital.

Step	Action	Running Balance
1	Borrow 1,000 ETH from Aave flash loan (0% interest)	+$1,000,000

Worked Example: Flash Loan Arbitrage

Scenario: Token X trades at $10.00 on DEX A and $10.50 on DEX B. You have zero starting capital.

Step	Action	Running Balance
1	Borrow 1,000 ETH from Aave flash loan (0% interest)	+$1,000,000
2	Buy 100,000 Token X on DEX A at $10.00 each	$0 cash + 100K tokens

Worked Example: Flash Loan Arbitrage

Scenario: Token X trades at $10.00 on DEX A and $10.50 on DEX B. You have zero starting capital.

Step	Action	Running Balance
1	Borrow 1,000 ETH from Aave flash loan (0% interest)	+$1,000,000
2	Buy 100,000 Token X on DEX A at $10.00 each	$0 cash + 100K tokens
3	Sell 100,000 Token X on DEX B at $10.50 each	+$1,050,000

Worked Example: Flash Loan Arbitrage

Scenario: Token X trades at $10.00 on DEX A and $10.50 on DEX B. You have zero starting capital.

Step	Action	Running Balance
1	Borrow 1,000 ETH from Aave flash loan (0% interest)	+$1,000,000
2	Buy 100,000 Token X on DEX A at $10.00 each	$0 cash + 100K tokens
3	Sell 100,000 Token X on DEX B at $10.50 each	+$1,050,000
4	Repay 1,000 ETH ($1,000,000) + 0.09% fee ($900)	+$49,100

Worked Example: Flash Loan Arbitrage

Scenario: Token X trades at $10.00 on DEX A and $10.50 on DEX B. You have zero starting capital.

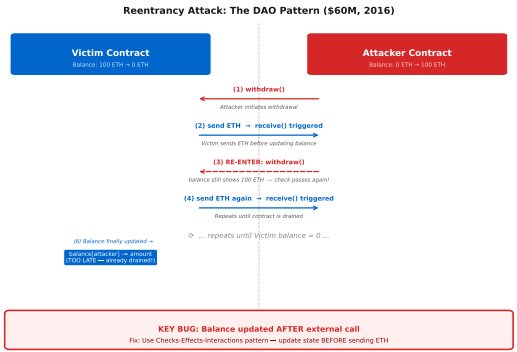
Step	Action	Running Balance
1	Borrow 1,000 ETH from Aave flash loan (0% interest)	+$1,000,000
2	Buy 100,000 Token X on DEX A at $10.00 each	$0 cash + 100K tokens
3	Sell 100,000 Token X on DEX B at $10.50 each	+$1,050,000
4	Repay 1,000 ETH ($1,000,000) + 0.09% fee ($900)	+$49,100
5	**Profit: $49,100 — in 13 seconds, with zero starting capital**	**+$49,100**

If any step fails (not enough liquidity, price moved, cannot repay), the *entire transaction reverts*. The blockchain acts as if nothing happened. The borrower loses only the gas fee (approximately $5).

This is risk-free arbitrage — a concept that was impossible in traditional finance because you needed capital to start.

Flash loans democratize arbitrage: you no longer need $1M in capital, just the knowledge to find price discrepancies

How a Reentrancy Attack Works



The bug: The victim contract sends ETH *before* updating its internal balance record.

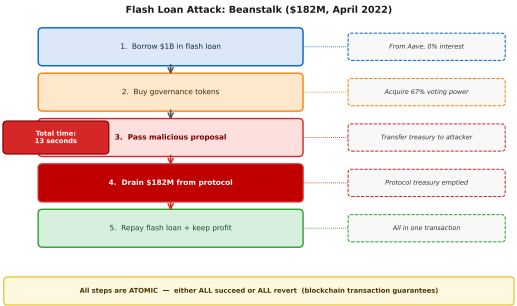
The exploit:

1. Attacker calls `withdraw()`
2. Victim sends ETH to attacker
3. Attacker's `receive()` function immediately calls `withdraw()` again
4. Victim checks balance — it was never updated, so the check passes
5. Repeat until the contract is drained

The fix: Update balances *before* sending ETH.
This is the **Checks-Effects-Interactions** pattern: check conditions, update state, then interact with external contracts.
The DAO hack (2016, $60M) used exactly this vulnerability.

The DAO hack was so severe it split Ethereum into two chains: Ethereum (rolled back the hack) and Ethereum Classic (kept it)

Flash Loan Attack Anatomy: Beanstalk ($182M)



April 17, 2022: The Beanstalk stablecoin protocol was drained.

Attack sequence:

1. Flash-borrow $1B in various tokens
2. Deposit tokens into Beanstalk to acquire governance voting power
3. Propose and immediately pass a malicious governance proposal
4. The proposal transfers all treasury funds to the attacker
5. Repay the flash loan
6. Net profit: $182M in one transaction

Root cause: Governance allowed instant vote execution. No timelock, no delay, no multi-signature requirement.

The attacker did not break any code — they used the governance system exactly as designed.

Beanstalk's governance had no timelock — a one-block governance attack was technically "legitimate" protocol usage

Oracle Manipulation: The Price Feed Problem

Oracle

An oracle is a service that provides external data (prices, weather, sports scores) to smart contracts. Since blockchains cannot access the outside world, oracles act as the bridge between on-chain logic and off-chain reality.

The problem: If an attacker manipulates the oracle, the smart contract acts on false information — and there is no way to "undo" it.

Spot price oracles report the current price on a single exchange. Easy to manipulate with a large trade.

TWAP oracles (Time-Weighted Average Price) average prices over a time window (e.g., 30 minutes). Much harder to manipulate because the attacker must sustain the false price.

Real attacks:

- **Cream Finance** ($130M, Oct 2021): Spot price oracle on a low-liquidity token
- **Harvest Finance** ($34M, Oct 2020): Rapid trades distorted Curve pool prices
- **Mango Markets** ($114M, Oct 2022): Manipulated spot oracle on Solana

In plain English: TWAP is like checking a stock price every hour for a day, instead of trusting a single quote from one trader.

Chainlink (decentralized oracle network) aggregates prices from dozens of sources — making single-source manipulation impractical

Security Audit: From Code to Confidence

No single tool catches every bug. A layered approach combines speed with thoroughness:

Layer	Method	Time	What It Catches
1	**Automated scanning** (Slither, Mythril)	Minutes	Known vulnerability patterns
2	**Fuzzing** (Echidna, Foundry)	Hours	Edge cases via random inputs
3	**Manual review** (human auditors)	Weeks	Logic errors, business logic flaws
4	**Formal verification** (Certora, Halmos)	Weeks	Mathematical proof of invariants
5	**Bug bounty** (Immunefi, Code4rena)	Ongoing	Unknown unknowns

Key insight: Layers 1–2 are fast but shallow. Layer 3 is deep but expensive. Layer 4 is mathematically rigorous but covers only specified properties. Layer 5 recruits the entire security community.

The most expensive audit is the one you skip: Beanstalk's governance bug would have been caught in Layer 3.

Top audit firms (Trail of Bits, OpenZeppelin, Consensys Diligence) charge $200K–$500K per audit — still cheaper than a $100M hack

Defense Mechanisms: Building Secure Protocols

Technical Defense Tools:

- **Checks-Effects-Interactions:** Always update state before calling external contracts
- **Reentrancy guards:** Mutex locks that prevent recursive calls
- **TWAP oracles:** Time-weighted prices resist single-block manipulation
- **Circuit breakers:** Automatic pause if TVL drops more than 25% in one hour
- **Timelocks:** Governance changes require 48–72 hour delay before execution

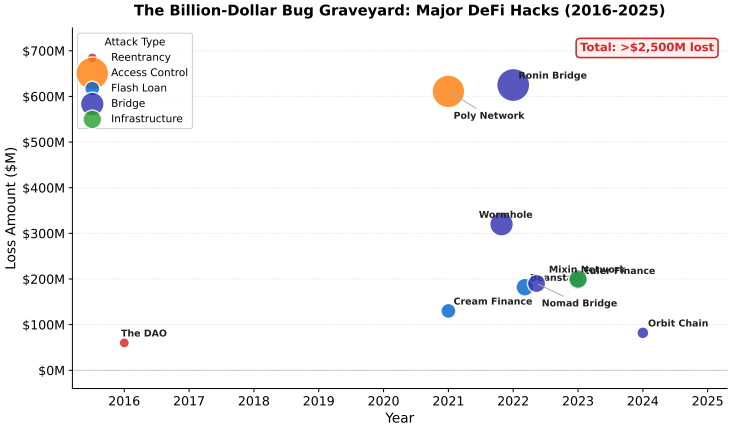
Operational Best Practices:

- **Multiple audits:** At least two independent firms review the same code
- **Gradual rollout:** Launch with TVL caps, raise limits over months
- **Bug bounties:** Pay up to $10M for critical vulnerabilities (cheaper than losing $320M)
- **Upgrade mechanisms:** Proxy contracts allow patching without redeployment
- **Multisig admin:** No single key can change critical parameters

The best defense is defense in depth: no single mechanism is sufficient, but layering them makes exploitation exponentially harder.

Uniswap V2 and V3 have held over $5B in TVL for years without a single exploit — proof that defense in depth works

The Billion-Dollar Bug Graveyard



- **What you see:** A timeline of major DeFi exploits from 2016 to 2025, with loss amounts
- **Key pattern:** Bridge attacks dominate the largest losses; flash loan attacks are the most frequent
- **Case studies:** The DAO ($60M, 2016, reentrancy), Ronin Bridge ($625M, 2022, compromised validators), Beanstalk ($182M, 2022, flash loan governance attack)

Total DeFi losses exceed $6B since 2016 — and these are only the publicly reported incidents

Bridge Attacks: The Weakest Link

Ronin Bridge
$625M — March 2022

5 of 9 validators compromised via social engineering (fake job offer to Axie Infinity developer).

Attackers forged withdrawal signatures. The theft went *unnoticed for 6 days*.

Root cause: Too few validators, weak key management.

Wormhole Bridge
$320M — February 2022

Signature verification bug allowed the attacker to mint 120,000 fake wrapped ETH on Solana without depositing real ETH on Ethereum.

Jump Trading (backer) covered the loss.

Root cause: Code bug in signature validation logic.

Nomad Bridge
$190M — August 2022

A routine upgrade introduced a bug that made every message valid. Once one person exploited it, 300+ addresses *copy-pasted* the exploit transaction.

Root cause: Initialization bug made zero-value proofs pass verification.

The "free money" hack.

Common thread: Bridges concentrate trust in a small set of validators or contracts. They are the single point of failure in a multi-chain world.

Bridge attacks accounted for over $2.5B in losses (2021–2023) — more than all other DeFi exploit categories combined

MEV: The Invisible Tax

MEV (Maximal Extractable Value) is the profit that block producers and searchers can extract by reordering, inserting, or censoring transactions within a block.

Sandwich attack example:

1. You submit a swap: 10 ETH for USDC
2. MEV bot sees your pending transaction
3. Bot buys before you (front-run), pushing the price up
4. Your swap executes at a worse price
5. Bot sells after you (back-run), capturing the difference

Your loss: approximately $50 per swap in invisible slippage.

Scale of the problem:

- Over $600M extracted from Ethereum users (2020–2024)
- Average MEV per block: $100–$500
- Sandwich attacks: 60% of all MEV

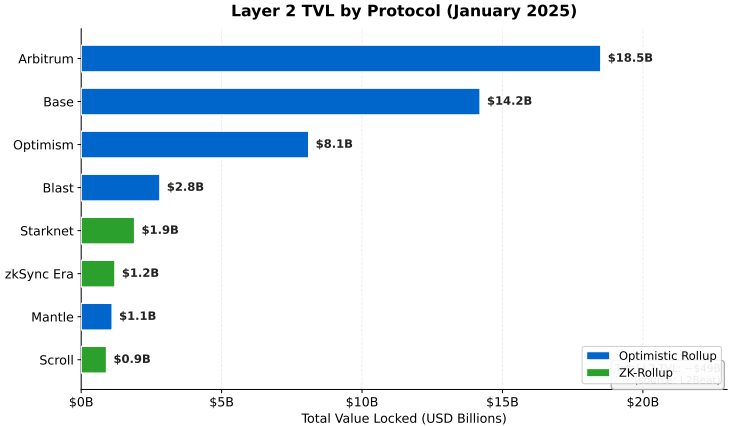
Why it is hard to stop:
MEV is not a bug — it is a *structural feature* of public mempools. Anyone can see your pending transaction before it is confirmed.

Partial solutions:

- Private mempools (Flashbots Protect)
- MEV-aware DEXs (CoW Swap)
- Order flow auctions

MEV is crypto's "hidden tax" — most users do not know they are paying it, but it costs DeFi users millions per month

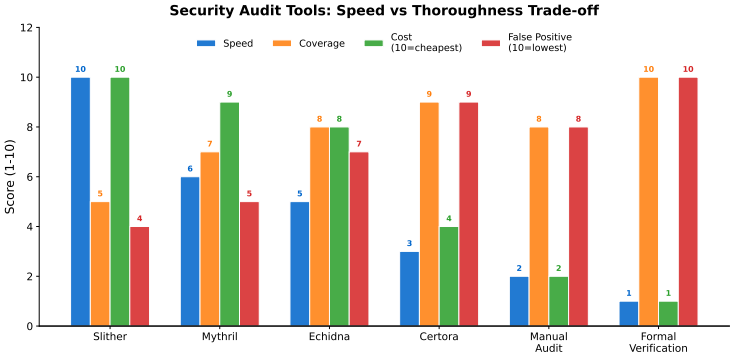
Layer 2 Market Landscape (2026)



- **What you see:** Total Value Locked across major Layer 2 networks, showing relative market share
- **Key pattern:** Arbitrum dominates with approximately 45% of L2 TVL, followed by Base (Coinbase) and Optimism. ZK-rollups (Starknet, zkSync) are growing fastest but still smaller.
- **Takeaway:** Total L2 TVL is approximately $49B (early 2026) — nearly half of Ethereum's DeFi TVL has migrated to Layer 2

Source: L2Beat.com, January 2026 — The L2 migration is accelerating — daily transactions on L2s now exceed Ethereum L1

The Security Industry: Tools and Players



- **What you see:** The smart contract security ecosystem mapped by speed versus thoroughness
- **Key pattern:** Automated tools are fast but shallow; manual audits and formal verification are slow but catch deeper bugs — no single tool catches everything
- **Takeaway:** Defense in depth requires combining automated scanning, fuzzing, human review, and bug bounties

The security audit market has grown to over $1B annually (2025) — a direct response to the scale of DeFi exploit losses

Bug Bounty Economics

Largest bounties paid:

Protocol	Bounty
Wormhole	$10,000,000
Aurora (NEAR)	$6,000,000
Polygon	$2,000,000
Optimism	$2,000,000
Arbitrum	$400,000

Total bounties paid (Immunefi):
Over $100M since 2020.

The economics are simple:

It is cheaper to pay a $10M bounty than lose $320M to a hack.

Bug bounties create a market for security:

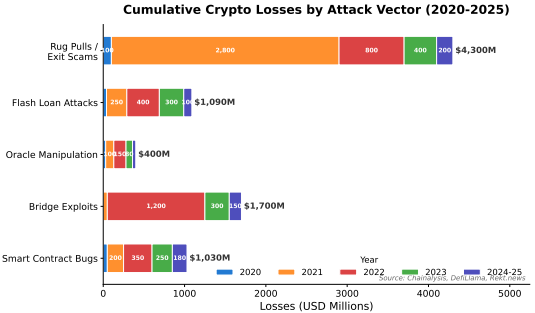
- White-hat hackers earn legitimate income
- Protocols get continuous security review
- The community benefits from safer infrastructure

The catch: Bounties only work if the payout is large enough to outweigh the temptation to exploit the bug yourself.

Rule of thumb: bounty should be at least 10% of potential exploit value.

Immunefi has facilitated over $100M in bounty payments — preventing an estimated $25B+ in potential losses

Attack Vector Losses: Who Pays?



Stakeholder impact:

Who	What They Lose
Users	Deposited funds
LPs	Liquidity positions
Protocol	TVL and reputation
Token holders	Token value crashes
Ecosystem	Adoption slows

There is no FDIC insurance in DeFi.
When a protocol is exploited, users rarely recover their funds. Some protocols (Wormhole, Euler) have repaid users from treasury or VC backing, but this is the exception.

The invisible cost: every major hack reduces public trust in the entire ecosystem.

Unlike traditional banking, DeFi has no deposit insurance — "code is law" means exploit losses are usually permanent

The Security-Innovation Trade-off

	Fast Innovation	Secure Development
Philosophy	"Move fast and break things"	"Move slowly and prove things"
Audit approach	Ship first, audit later	Audit first, ship later
Time to market	Weeks	Months
Cost	Low upfront, high risk	High upfront, low risk
Examples	Many DeFi forks (2020–2021)	Uniswap, Aave, MakerDAO
Outcome	Some succeeded, many were hacked	Survived multiple market cycles

The middle path — Progressive Security:

- Launch with low TVL caps and circuit breakers
- Run automated scanning and fuzzing from day one
- Commission full audit before raising TVL limits
- Maintain an ongoing bug bounty for unknown unknowns
- Use timelocks on all governance changes

The protocols that survived 2022–2023 (Aave, Uniswap, Compound) all followed progressive security.

"Move fast and break things" works for social media — in DeFi, breaking things means losing real money irreversibly

The 5-Layer Security Framework

A systematic checklist for evaluating any protocol's security posture:

1. **Code Layer:** Has the smart contract been audited by at least two independent firms? Are automated scans (Slither, Mythril) clean? Has fuzzing been performed?
2. **Protocol Layer:** Are economic invariants formally specified? Does the protocol use TWAP oracles or Chainlink (not spot prices)? Are there flash loan protections on governance?
3. **Infrastructure Layer:** How secure are the bridges? Are admin keys stored in hardware wallets? Is there a sequencer fallback for L2s?
4. **Governance Layer:** Is there a timelock (48–72 hours minimum)? Does governance require multisig approval? Is voting power decentralized (no whale dominance)?
5. **Ecosystem Layer:** Is there a bug bounty program with meaningful payouts? Does the protocol carry smart contract insurance (Nexus Mutual, InsurAce)? Are circuit breakers in place?

If a protocol fails on any layer, the layers below it are undermined — like a building with a weak foundation.

Use this framework to evaluate any DeFi protocol before depositing funds — check L2Beat.com for Layer 2 risk assessments

Evaluating L2 and Protocol Risk

Evaluating Layer 2 Networks:

- **Security model:** Does the rollup post fraud/validity proofs to L1, or does it rely on its own validator set?
- **Sequencer centralization:** Is there a single sequencer? What happens if it goes offline?
- **Withdrawal guarantees:** Can users force-withdraw to L1 without sequencer cooperation?
- **Track record:** How long has it operated without incident? What is its TVL stability?

Check L2Beat.com — it rates every L2 on these criteria.

Evaluating DeFi Protocols:

- **Audit history:** How many audits? By which firms? Were findings fixed?
- **TVL stability:** Has TVL been consistent, or does it spike and crash?
- **Governance decentralization:** Can one whale or flash loan control voting?
- **Incident response:** Has the team handled a crisis before? How fast?

Red flags:

- No audit or single audit only
- Anonymous team with no track record
- No timelock on governance
- No bug bounty program

Due diligence in DeFi is your responsibility — there is no regulator, no deposit insurance, no customer support hotline

What Comes Next

Emerging trends (2025–2027):

- **Formal verification becoming standard:** Mathematical proofs of contract correctness, not just audits
- **ZK-everything:** Zero-knowledge proofs for privacy, scaling, identity, and compliance simultaneously
- **Account abstraction (ERC-4337):** Smart wallets with social recovery, gas sponsorship, and session keys — making crypto usable for normal people
- **Cross-chain interoperability:** Standards for secure message passing between chains (replacing vulnerable bridges)

The big picture:

The industry is converging on a layered architecture: Ethereum as the settlement layer, rollups for execution, and ZK proofs for everything else.

Challenge for you:

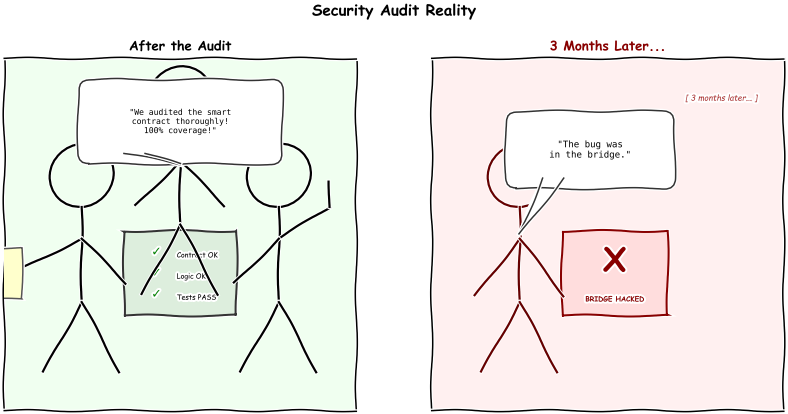
Pick one Layer 2 protocol. Go to `L2Beat.com` and read its risk assessment. Answer these questions:

1. What is its security model?
2. Is the sequencer centralized?
3. Can you force-withdraw to L1?
4. What is its biggest vulnerability?

Bring your findings to the next class.

The next 3 years will determine whether DeFi becomes mainstream infrastructure or remains a niche experiment

[Cartoon] One Last Thought...



You can audit the contract, but not the infrastructure around it.

"Security is not a product. It is a process." — Bruce Schneier

Every protocol that survived the 2022 bear market had one thing in common: layered security, not a single silver bullet

Key Takeaways

Seven ideas to remember from today:

1. **The scalability trilemma forces trade-offs** — Layer 2 networks try to achieve all three properties by processing off-chain while anchoring security to L1.
2. **Rollups inherit L1 security** by posting compressed transaction data and fraud/validity proofs to Ethereum.
3. **Dencun (EIP-4844) reduced L2 fees by 98%** via blob transactions — temporary data storage that is sufficient for rollup security.
4. **Flash loans enable zero-capital attacks** — any amount, one transaction, 13 seconds. If repayment fails, everything reverts.
5. **Reentrancy, oracle manipulation, and bridge exploits** account for over $3B in cumulative losses since 2016.
6. **No single security tool catches everything** — defense requires layering automated scanning, human audits, formal verification, and bug bounties.
7. **The 5-Layer Security Framework** (Code, Protocol, Infrastructure, Governance, Ecosystem) provides a systematic method for evaluating any protocol.

Review question: Can you apply the 5-Layer Security Framework to evaluate Arbitrum's risk profile? Check L2Beat.com for data.

Summary and Next Module Preview

Summary:

The tension between innovation and security defines the frontier of blockchain development. Layer 2 solutions solve the scalability trilemma at the cost of new infrastructure risks. Flash loans democratize capital access but enable novel attack vectors. Smart contract security has evolved from "ship and pray" to a multi-layered discipline combining automated tools, human expertise, and economic incentives. The protocols that survive are those that treat security as a process, not a checkbox.

Key Vocabulary:

1. Rollup
2. Flash Loan
3. Reentrancy
4. MEV
5. TWAP
6. Bug Bounty

Next: Module G — Regulation & Future

How the legal and regulatory landscape shapes what blockchain can and cannot do:

- MiCA regulation (EU, fully effective Dec 2024)
- SEC enforcement actions and legal precedents
- CBDC design and central bank digital currencies
- The future of decentralized governance

Connection: Every security exploit we studied today is also a *regulatory event*. Bridge hacks trigger enforcement actions. Flash loan attacks raise questions about market manipulation law. MEV challenges the legal definition of "fair" markets.

Module G asks: who makes the rules when the system is designed to have no ruler?

Key vocabulary: Master these 6 terms — they connect Layer 2, DeFi security, and regulation into a unified framework