Adaptive Counter-Attack Synthesis for Mitigation of Onchain Exploits

20th SoCal PLS: Programming Languages and Systems

Hanzhi Liu[1], Yanju Chen[2], Jiaming Shan[1], Jiaxin Song[3], Chaofan Shou[4], **Hongbo Wen**[1], Yu Feng[1]

[1]UCSB [2]UCSD [3]UIUC [4]Fuzzland

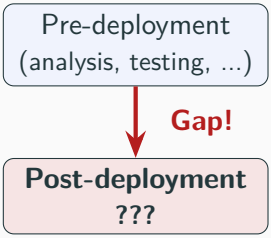
The Problem: Smart Contract Exploits

DeFi: \$90B+ locked in immutable smart contracts.

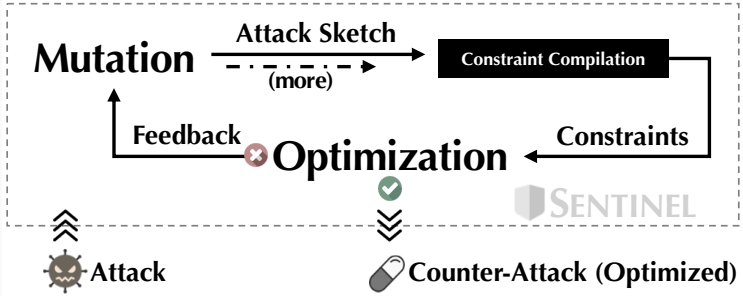
Existing defenses are all **pre-deployment**:

- Static analysis, fuzzing, symbolic execution, verification
- Cannot catch everything (Rice's Theorem)

Once deployed, bugs are irreversible.
Attackers exploit them in **seconds**.
$\Rightarrow$ billions of dollars lost.



Sentinel: Real-Time Counter-Attack Synthesis



Phase 1: Attack Sketch Inference

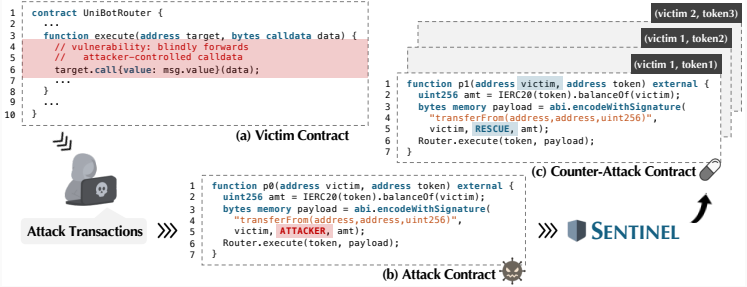
- Trace exploit → symbolic sketch
- Extract attacker's financial goal

Phase 2: Counter-Attack Optimization

- Mutate sketch, compile to SMT
- Maximize asset recovery → deploy

Deployed inside **L2 sequencers** ⇒ guaranteed front-running.

Example: UniBot Exploit



Bug: Router forwards calldata with *no auth check*.

Attack: `transferFrom(victim, ATK, amt)`

Prior work (STING): Rewrite recipient $\rightarrow$ saves *one* victim.

Sentinel: Generalize to *all* victims $\times$ tokens in one tx.

Key Technical Ideas

1. DeFi-Aware IR $\mathcal{L}_c$

DeFi ops as first-class primitives:

$\sigma \in \{\text{transfer}, \text{swap}, \text{borrow}, \text{mint}, \ldots\}$

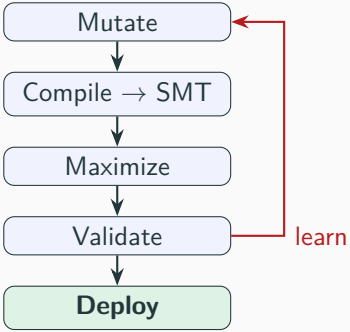
$\Rightarrow$ compact SMT constraints

2. Mutation Language $\mathcal{L}_m$

6 operators: insert, replace, symbolize, split, merge, map/acc

map/acc: batchify across all victims

3. Conflict-Driven Synthesis



Failed $\rightarrow$ root cause $\rightarrow$ prune future search

Evaluation: 33 Real-World Exploits

Victim	STING	Sentinel	Imp.
Discover	100K	19.2M	192×
ShadowFi	3.5 WBNB	424 WBNB	121×
NOVO	1.3 WBNB	100 WBNB	80×
OneRing	16.9K	1.25M	74×
Unibot	84K	729K	8.7×
Balancer V2	68.3M	80.7M	1.18×
Avg (33)	**$3.0M**	**$5.1M**	**1.67×**

Headlines:

- **33/33** benchmarks solved
- Avg. synthesis: **64 ms**
- Up to **192**× more assets saved
- Est. **$167M** preserved

Ablation vs. Halmos:

- Halmos: 3/33 solved, avg 504s
- SENTINEL: 33/33, avg 0.06s
- ∼8,000× faster

Live Deployment: 13 Zero-Day Exploits Mitigated

Deployed with an Ethereum L2 partner, June–September 2025.

13 previously unknown exploits neutralized

Protocol X (Balancer V2 fork)

- Rounding flaw → rescued $22K
- Original Balancer: $128M lost

Protocol Y (Trading bot)

- Missing auth → generalized to **57 tokens** × **68 victims**

Protocol Z (Lending)

- Reentrancy → rescued **$360K+**

Conclusion

Sentinel = real-time, automated counter-attack synthesis for on-chain exploits.

What we showed:

- DeFi-aware IR + mutation DSL
- Conflict-driven synthesis in **64 ms**
- 33/33 exploits, **$167M** preserved
- 13 live zero-days mitigated

Future work:

- Layer-1 & MEV relayer integration
- Multi-VM (Solana, Move, ...)
- Cross-contract invariant reasoning

Thank you! Questions?