Houston:

Real-Time Anomaly Detection of Attacks Against Ethereum DeFi Protocols

Dongyu Meng*, Fabio Gritti*, Robert McLaughlin, Nicola Ruaro,
Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna

UC SANTA BARBARA SECLAB

NDSS SYMPOSIUM/2026

Decentralized Finance (DeFi)

Financial services implemented on-chain as composable smart contracts

Includes:

- Decentralized exchanges (DEX)
- Lending platforms
- Stablecoins
- and more

~$100B in Total Value Locked (TVL) and growing

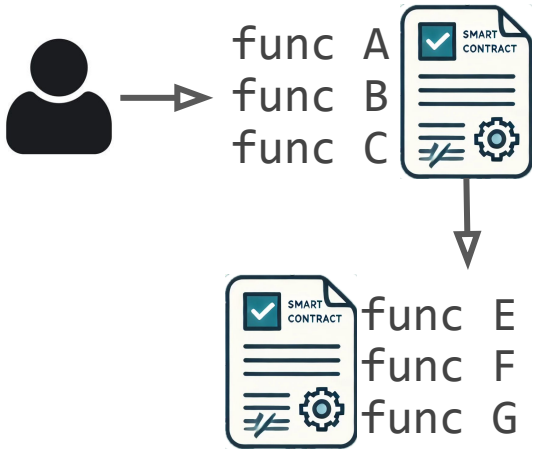
Smart contract:

- Program deployed to a certain address on-chain

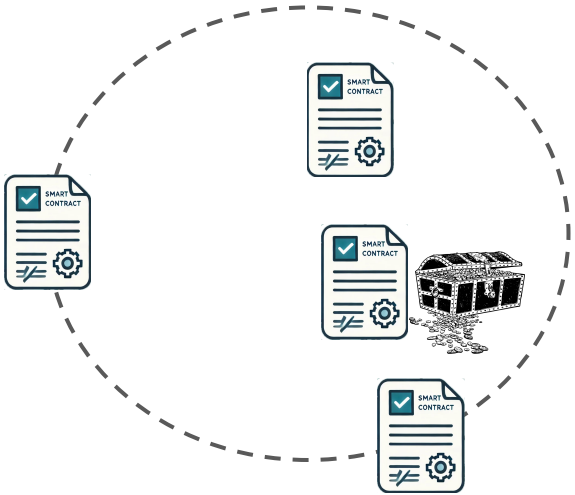

```
func A
func B
func C
```

Smart contract:

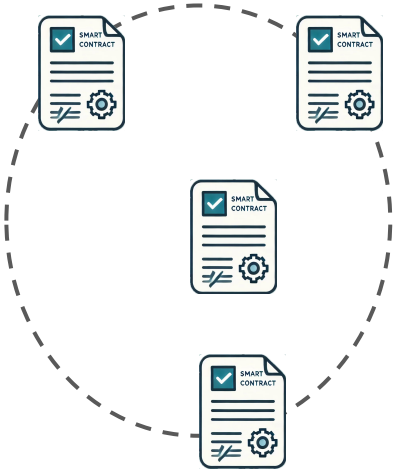
- Program deployed to a certain address on-chain
- Interact with users and other smart contracts via calls



DeFi Protocols



E.g., lending protocol



DeFi under Attack

Total losses from crypto hacks in 2023 amounted to

$3.3 billion

$2.5B Lost in 2025

Security In Crypto

FBI – Federal Bureau of Investigation

February 27, 2025 ·

North Korean cyber actors have stolen approximately $1.5 billion in Ethereum from Bybit

Cryptocurrency Sector Faces 16 Hacks in January 2026, Losses Reach $86.01 Million

~$100B in TVL is at stake

DeFi Protocol Vulnerabilities

Malicious transactions exploit vulnerabilities in the protocol.

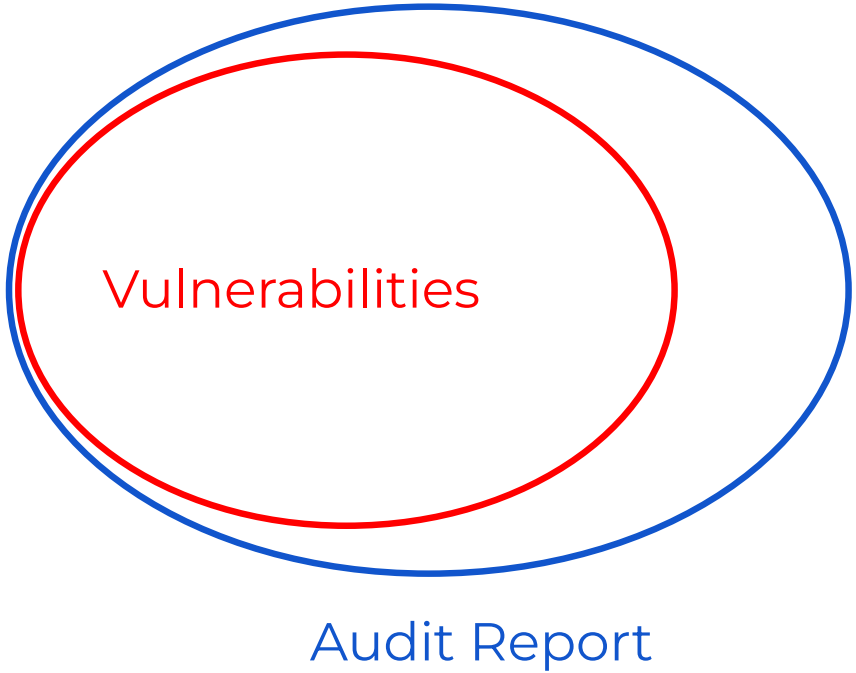
- Reentrancy
- Missing access control
- Precision loss / rounding errors
- Price oracle manipulation
- Storage layout collision
- ……

Auditing

Before deployment
Industry common practice

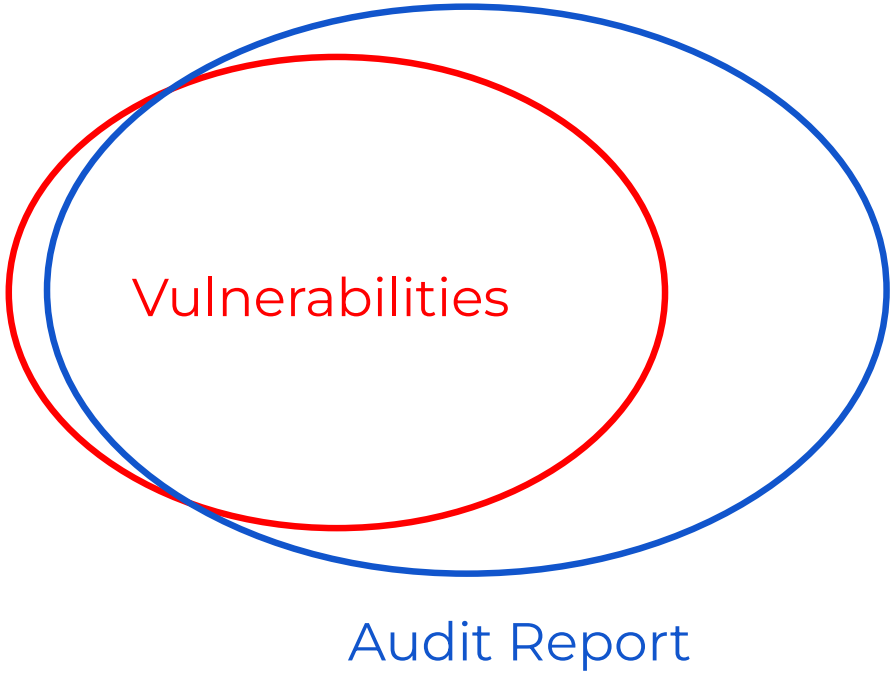
- Manual auditing
- Automated tools (e.g., fuzzing)
- Formal verification
- LLM-assisted auditing

Auditing



Many major hacks happened on audited protocols.

Auditing



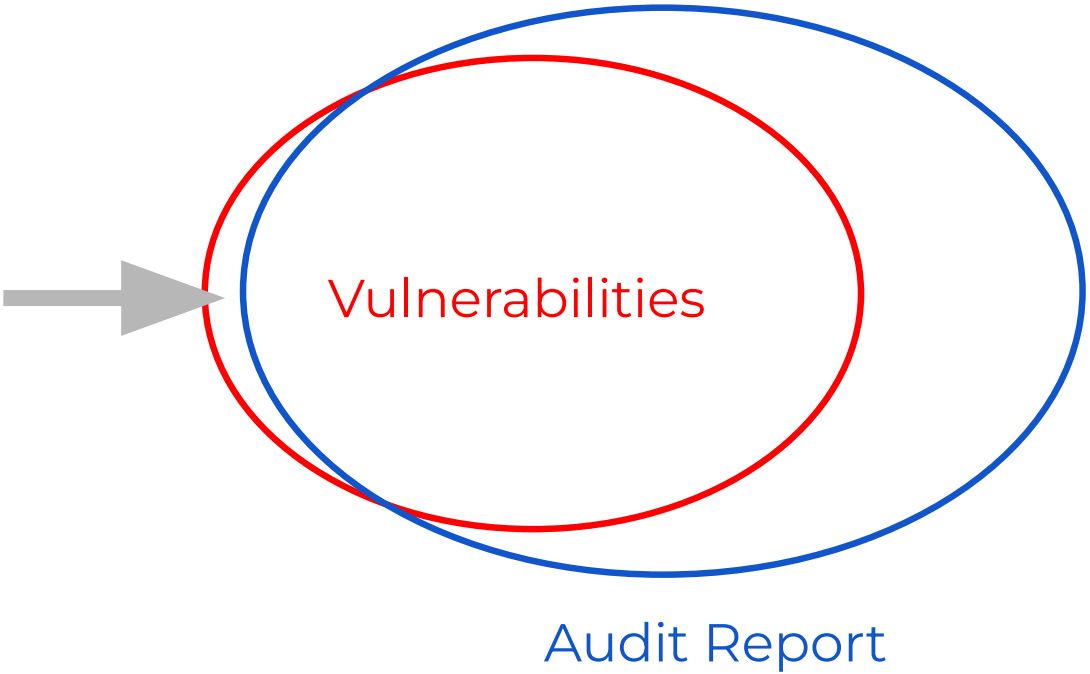
Auditing

Monitoring

Monitoring

After deployment
Detect suspicious transactions
Intervene upon detection ASAP
Even stop tx before execution
(e.g., while in mempool)

Monitoring



Challenges

Fast - within block generation interval (12s for Ethereum)
Agnostic - detect anomalous behavior regardless of vulnerability types
Low-noise - minimizes alert fatigue

Houston

Light-weight, off-chain, streaming anomaly detection on protocol usage.

For each protocol, Houston maintains an evolving behavior specification learned from transaction traces.

- Infer initial specification from historical interactions (if any)
- Flag abnormal transaction upon violation
- Triage alerts, expand allowed behaviors on False Positives
- Refine behavior modeling as benign transactions accumulate

Houston


DEFI PROTOCOL
CAUTION: LEARNED FRAGILITY
DELICATE
HANDLE WITH CARE
LEARNING MATERIALS
PLEASE BE CAREFUL
OPENING SOON

A Malicious TX in a China Shop

Houston: Two Models

Captures program specifications in **control- and data- flow** representations:

- **Control flow: Interaction Model**

Interaction Model

- **Heuristic normalization** $\mathscr{A}$ that compresses complex call traces into compact sequences $\mathscr{A}(\texttt{raw trace})$.
  - focuses on externally initiated state-changing interactions.
  - protocol-specific, guided by its implementation.
  - reduces modeling space significantly.
- **Database** $\mathscr{D}$ of legitimate (seen) compressed call sequences.

Interaction Model

Euler protocol, hacked in 2023

Decentralized lending protocol.
Txs can be very complicated and the modeling space is huge.

Interaction Model

Euler protocol, hacked in 2023

Attack transaction `raw trace`:

- **150+** function invocations.

Interaction Model

Euler protocol, hacked in 2023

Attack transaction `raw trace`:

- **150+** function invocations.

After heuristic normalization:

- $\mathscr{A}$`(raw trace)` only has **8** function invocations.
- Exact steps as described in writeups!

Interaction Model

Euler protocol, hacked in 2023

Attack transaction `raw trace`:

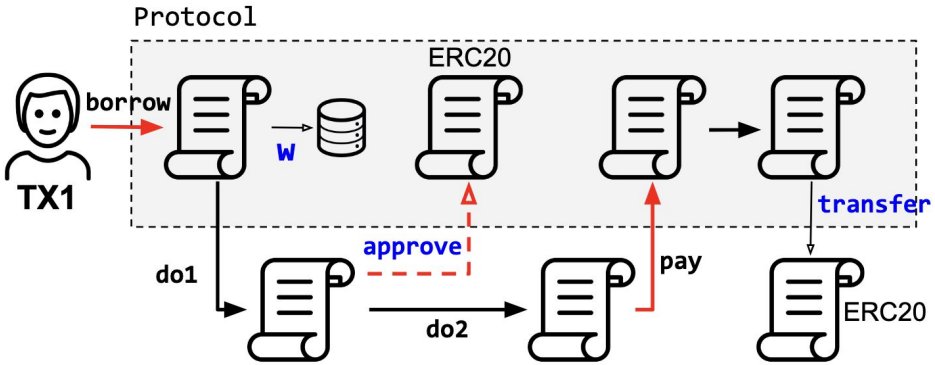
- **150+** function invocations.

After heuristic normalization:

- $\mathscr{A}$(`raw trace`) only has **8** function invocations.
- Exact steps as described in writeups!

$\mathscr{A}$(`raw trace`) exhibited a novel sequence absent from database $\mathscr{D}$, triggering an alert.

Interaction Model



`fingerprint` $_{TX1}$ `= [borrow, pay]`

1. Call Direction Identification
   - only keep incoming calls
2. Critical Call Identification
   - triggers storage changes
   - initiates predefined functions
     - ERC20 operations
3. Direct Token Operation Filtering
4. Interaction Pattern Identification

See details in the paper

Houston: Two Models

Captures program specifications in **control- and data- flow** representations:

- **Control flow**: **Interaction Model**
- **Data flow**: **Invariant Model**

Invariant Model

Likely Invariants

Properties that are dynamically inferred from values observed during multiple program executions.

E.g., State *variable X has never been smaller than function argument Y at EXIT of F.*

Invariant Model

- **Incremental likely invariant miner $\mathcal{M}$** that given
  - variable (pairs) `vars`
  - likely invariant template `T`
  - current likely invariant over `vars`: `INV` (can be empty or violated)
  - new transaction trace `d`

  $\mathcal{M}$(`vars, T, d, INV`) $\mapsto$ $\texttt{INV}_{\texttt{new}}$
- **Database $\mathscr{D}$** of established likely invariants for the protocol.

Invariant Model

Cover Protocol, hacked in 2020

Mined likely invariant in Database $\mathscr{D}$:
`_amount < _totalSupply` for function `mint` entrance.

Historically, during minting, `_amount` was always less than `_totalSupply`.

Invariant Model

Cover Protocol, hacked in 2020

Mined likely invariant in Database $\mathscr{D}$:
`_amount < _totalSupply` for function `mint` entrance.

However, in the attack transaction, due to a business logic bug in the protocol that mis-calculates the reward amount for staking users, over 40 quintillion COVER tokens were minted, causing the invariant to be violated (`_amount > _totalSupply`).

Houston therefore raised an alert.

Invariant Model

Local properties (function entrance/exit)
Critical variables (function arguments, contract storage variables)
Simple, generic invariant types

- (unary, integer) var always not zero
- (unary, bytes/addr) var drawn from a limited set of possible values
- (binary, integer) var1 and var2 have ordering relations (=,≥,≤)
- (binary, string/addr/bytes,) var1 == var2

Homebrewed inference engine for incremental reference.

See details in the paper

Evaluation Dataset

115 Ethereum DeFi Incidents

- `8.6M` transactions
- Jan 2020 to Sep 2024
- Covers a wide range of vulnerability types

Evaluation Results

True positive: **94.8%** (109/115)
False positive: **0.16%** (0.4 FP per protocol day)

All transactions finished end-to-end detection within block time.

Live Evaluation

We deployed Houston on live Ethereum traffic, monitoring 20 DeFi protocols for 40 days.

Performance

- Easily handled on commodity hardware
- Light-weight runtime overhead

Outcome

- No attacks observed during the period (also no attack reported)
- 0.07% false positive rate

Conclusion

Houston is an anomaly detection system for DeFi protocols

- models protocol behaviors with control- and data-flow properties
- Real-time, vulnerability-type-agnostic, low-noise, and explainable

Evaluated on 115 real-world Ethereum attack incidents

- 94.8% detection rate, 0.16% false positive rate, outperforms SotA

Live evaluation confirms the practicality of the system

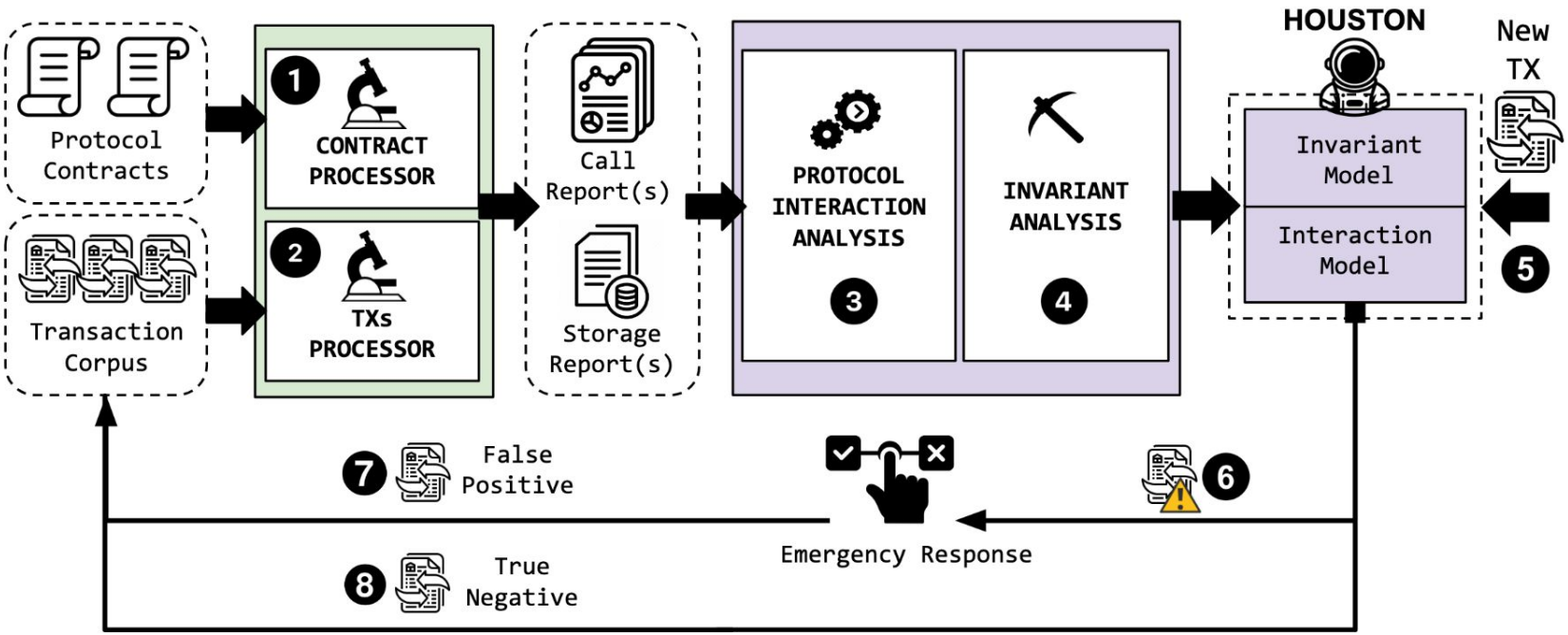
Conclusion

Houston is an anomaly detection system for DeFi protocols
- models protocol behaviors with control- and data-flow properties
- Real-time, vulnerability-type-agnostic, low-noise, and explainable

Evaluated on 115 real-world Ethereum attack incidents
- 94.8% detection rate, 0.16% false positive rate, outperforms SotA

Live evaluation confirms the practicality of the system

Thank you! Questions?


Protocol Contracts
Transaction Corpus
1
CONTRACT PROCESSOR
2
TXs PROCESSOR
Call Report(s)
Storage Report(s)
PROTOCOL INTERACTION ANALYSIS
3
INVARIANT ANALYSIS
4
HOUSTON
Invariant Model
Interaction Model
New TX
5
6
Emergency Response
7
False Positive
8
True Negative

Results: False Negatives

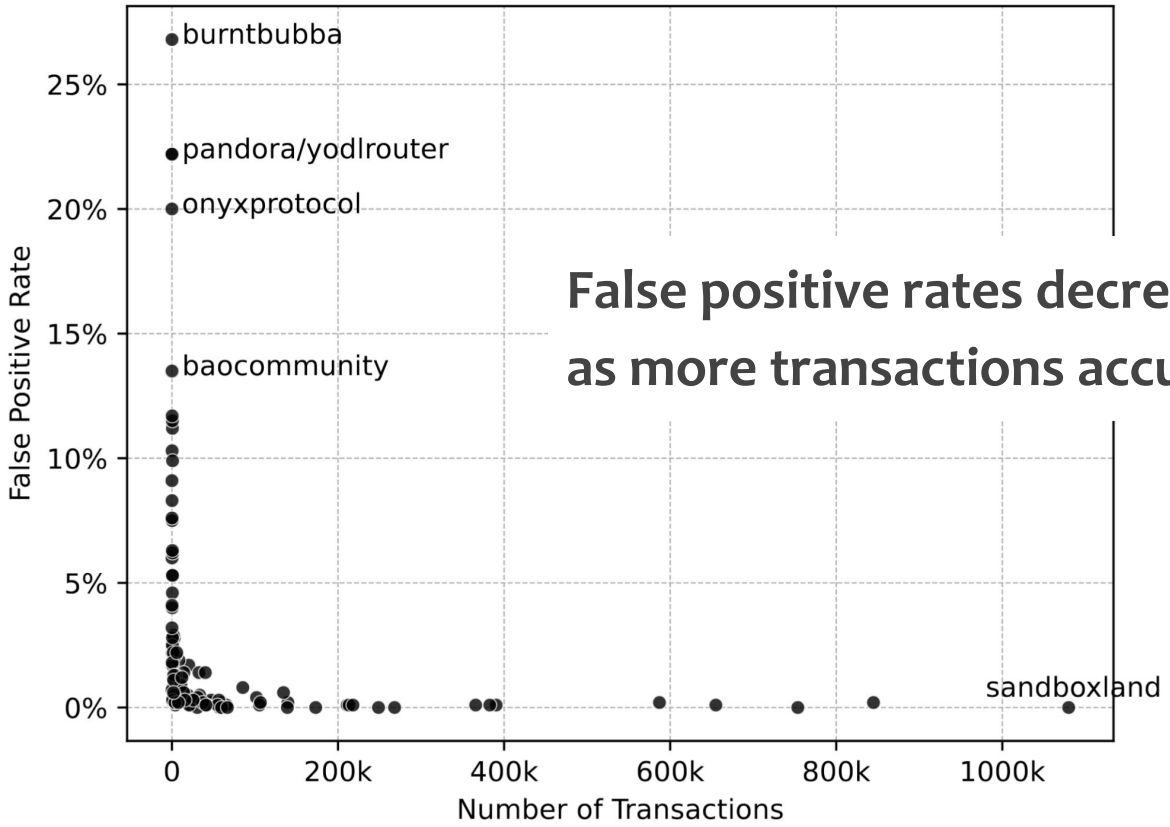
Reasons for False Negatives:

- Unconventional storage layout (e.g., dexible, dynamic layout)
- Unconventional contract code execution (e.g., nomad JUMP)
- Insufficient transaction tistory (e.g., azukiDAO, hours)
- Complex invariant types (bzx, conjunction of two binary invariants, once)

Results: False Positives

- Operation repetition and permutation
  - e.g., 5 deposits and 1 withdraw
- Historical data availability
  - e.g., new useage, new contracts
- Likely invariant quality

Evaluation: Attack Detection



False positive rates decrease rapidly as more transactions accumulate

Comparison with SotA

BlockGPT:

Overall, our system detects 27 attacks against these 28 protocols, while BlockGPT only detects 15.

Overall, BlockGPT achieves a false positive rate of less than 10% for 58% of the protocols in its scope. As a comparison, HOUSTON has a false positive rate of less than 10% for 98% protocols. Notably, HOUSTON achieves a false positive rate of 1% or less for 67% of the protocols in our dataset.

DeFi Ranger:

74.8% incidents out of scope

Comparison with SotA

TXSPECTOR:
TPR: 21.7% - Houston: 94.8%

APE:
On APE's dataset:
TP: 4/20 - Houston: 19/20
FPR: 0.15% - Houston: 0.10%

Length Baseline:
TPR: 40%

Limitations

- Prevention requires visibility of transaction.
- Requires source code of the monitored protocol.
- Refine speed and false positive rate for mass adoption.
- Dynamic adversary.

Hyperparams

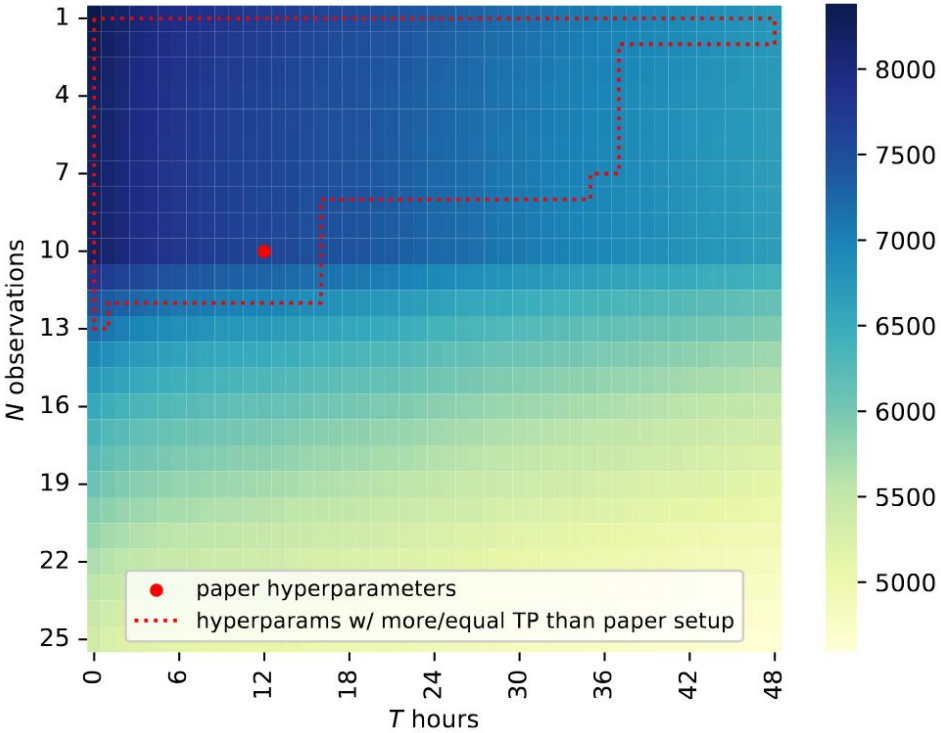
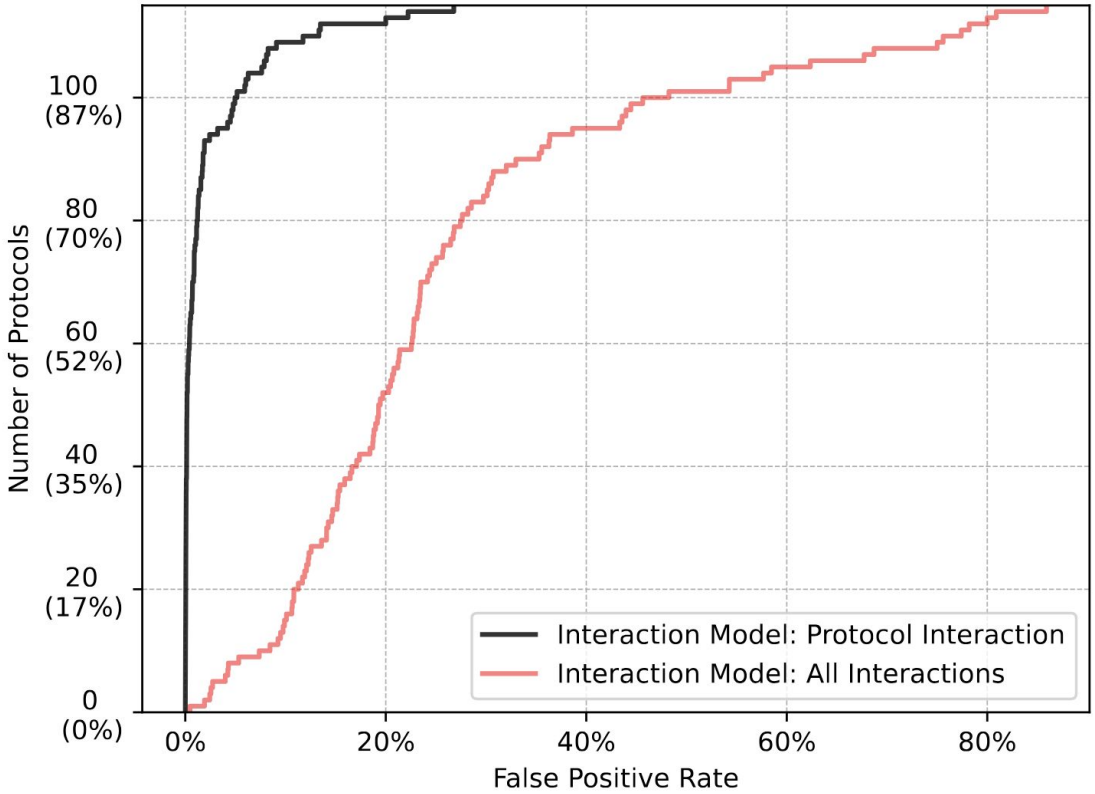


Fig. 8: Total number of false positives of the *Invariant Model* across all protocols with different hyperparameter setups. The paper setup does not overfit to the dataset.

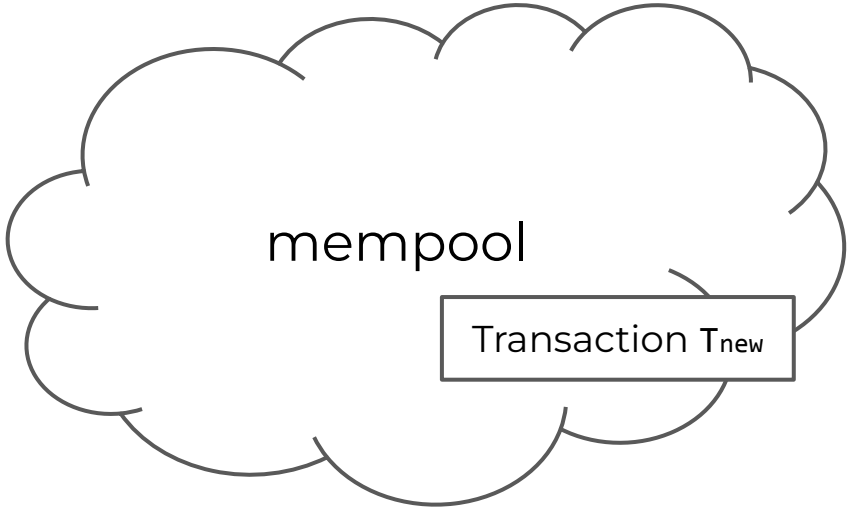
Ablation: Interaction



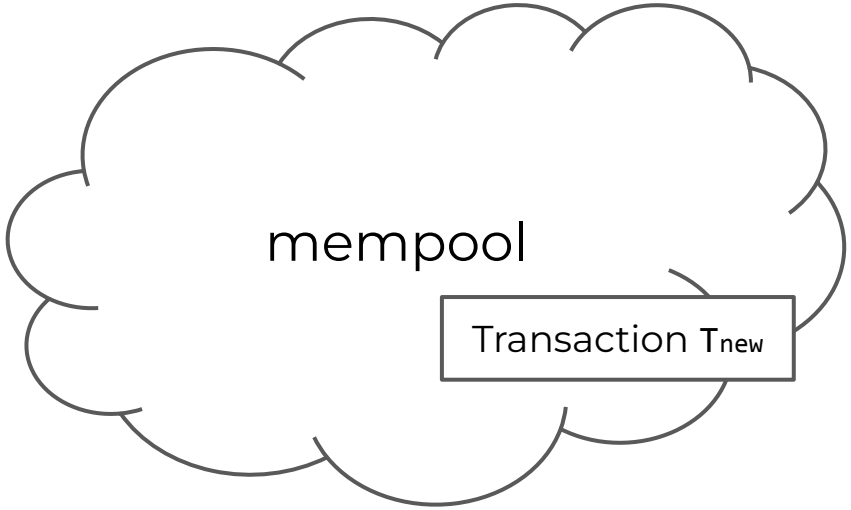
192 times higher FPR using raw call trace

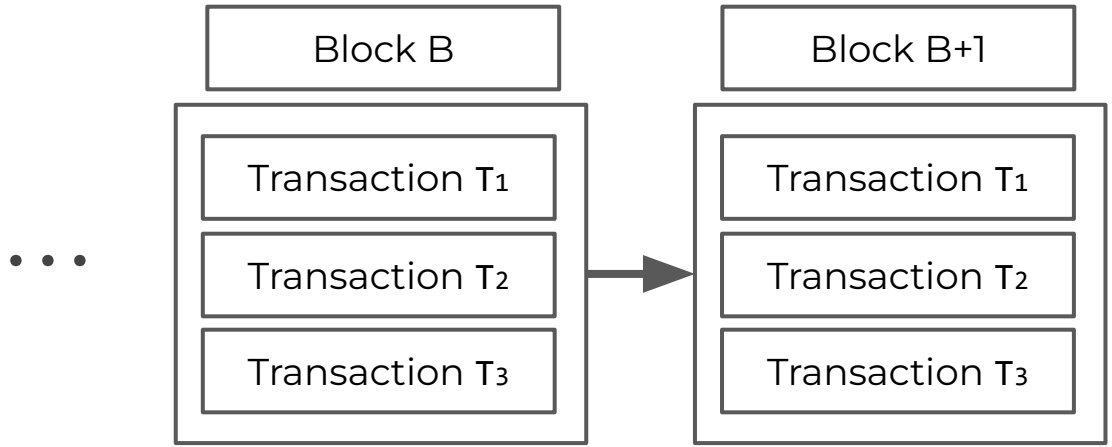
Thank You!

Transaction

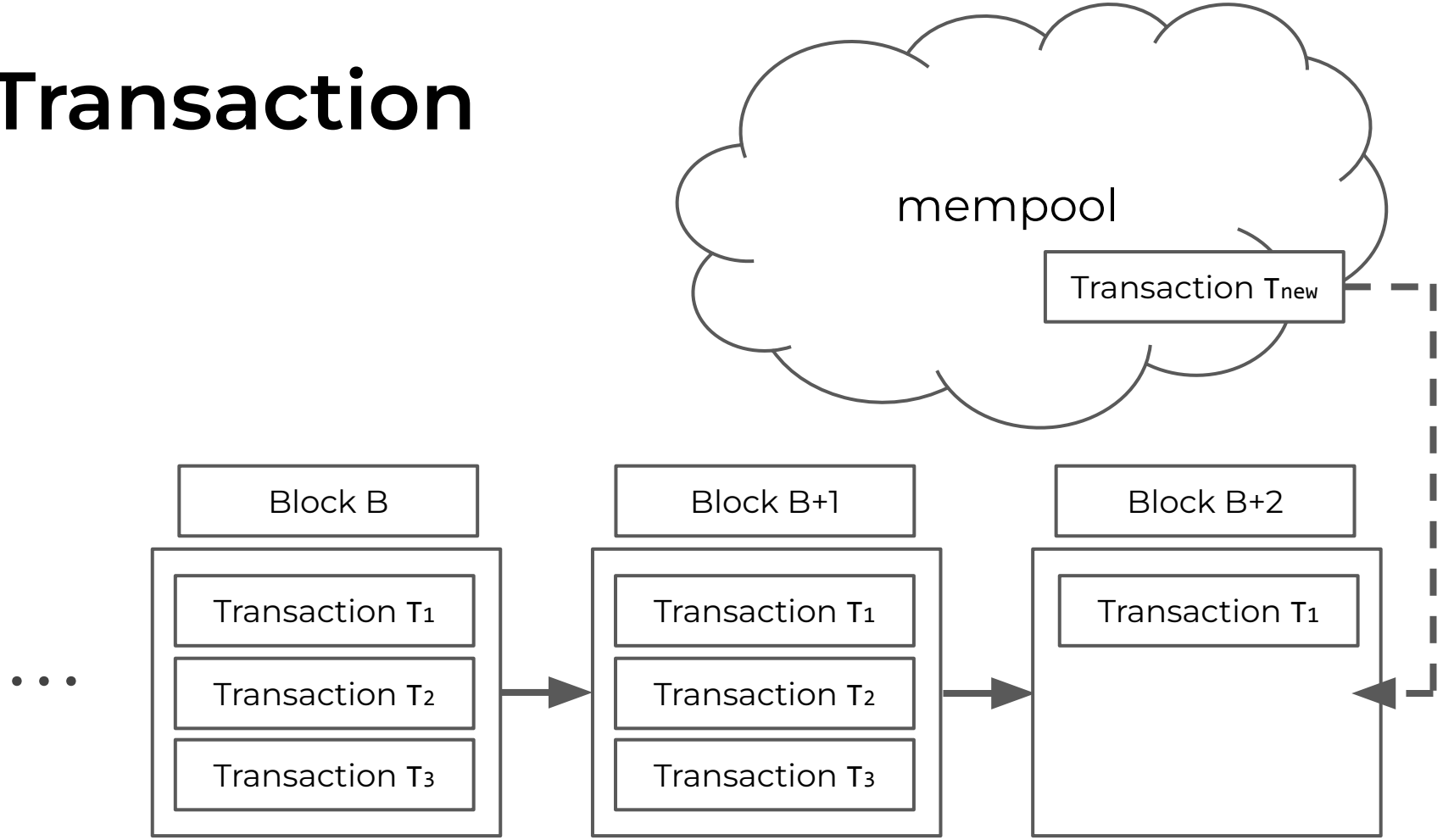


Transaction

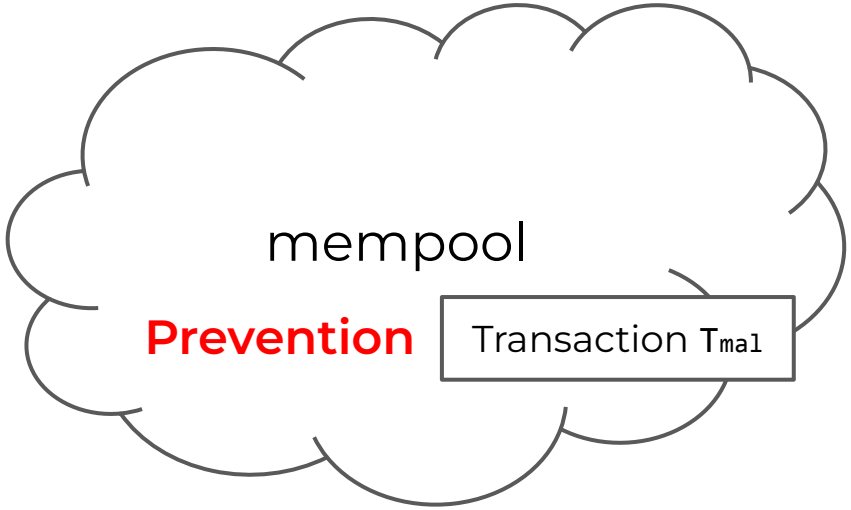


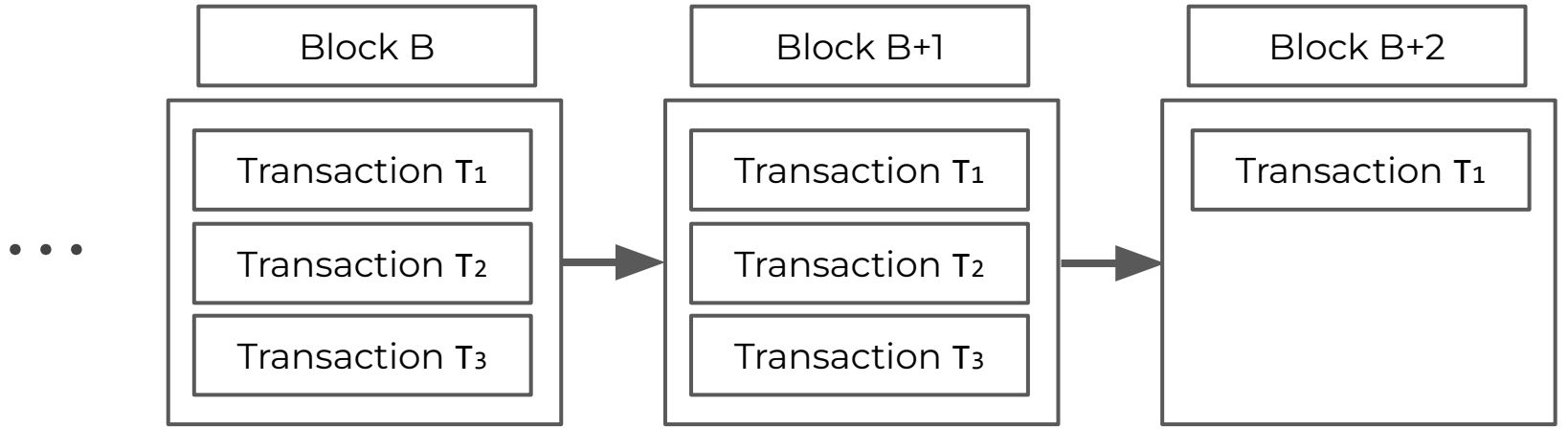


Transaction

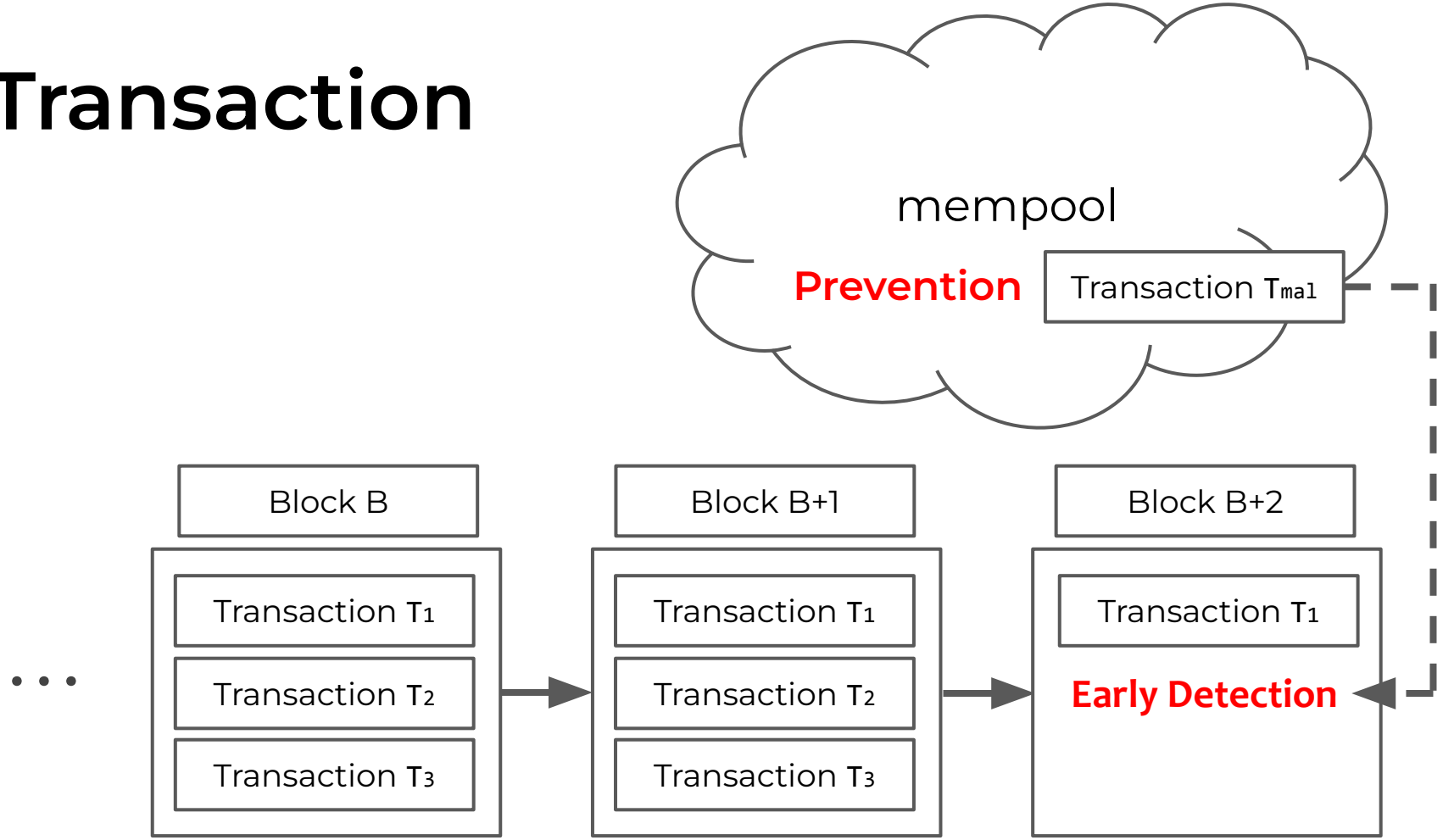


Transaction





Transaction



Evaluation Scheme

Replay the protocol timeline assuming Houston is deployed at genesis (no prior history) until the first attack occurs.

- True Positive: alert on the labeled attack transaction
- False Positive: alert on any non-attack transaction

Evaluation: Attack Detection

True positive: **94.8%** (109/115)
False positive: **0.16%**

End-to-end detection time (in second, need to be < 12s):

Max	7.119
99%	1.029
Median	0.269

Evaluation: Attack Detection

True positive: **94.8%** (109/115)

For the true positives cases, the provided evidence is:

- **Causal: 52%** (57/109)
  - pinpoints the root cause of the vulnerability
- **Indicative: 39%** (43/109)
  - reflects a consequence of the attacker behavior

More Evaluation

Live Performance Evaluation on real Ethereum traffic:

- Light computation and memory footprint.

Additional experiments on other **EVM-compatible chains** (BSC):

- Generalize across chains.

.....

Please check out the paper for details.