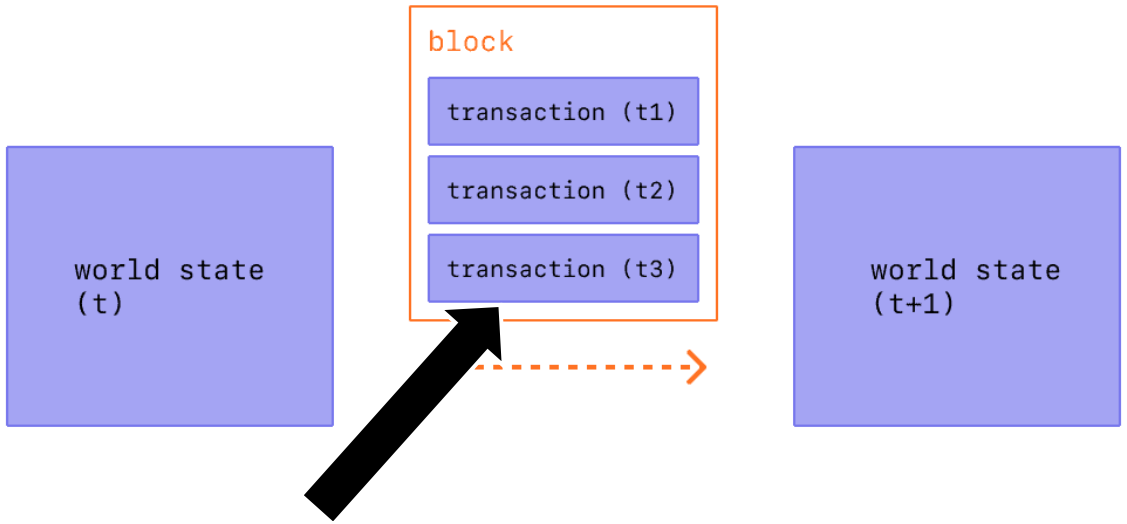
The Blockchain Imitation Game

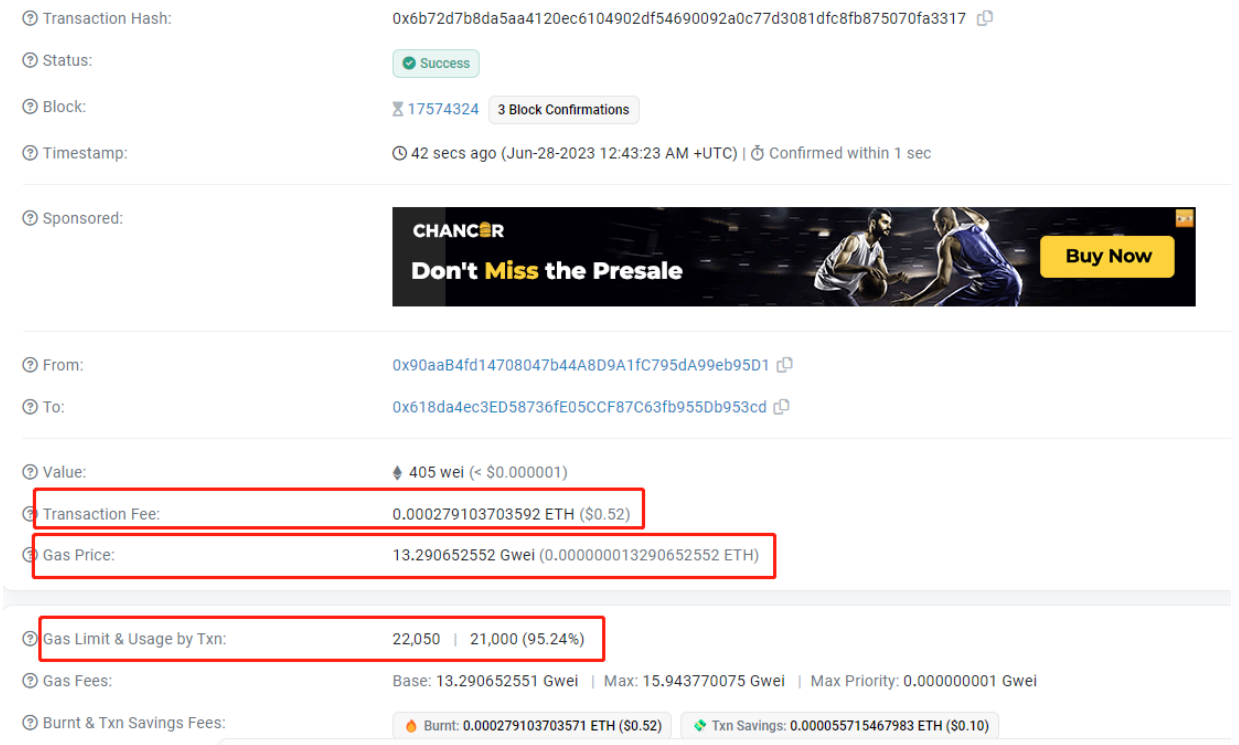
Kaihua Qin , Stefanos Chaliasos , Liyi Zhou ,
Benjamin Livshits , Dawn Song, Arthur Gervais

2023 USENIX SECURITY SYMPOSIUM

Background

- **Smart Contract**： implement a set of rules for managing digital assets in Ethereum accounts.





Transaction fee = Gas Price * Gas

Background

Taint Analysis

tracks information flow originating from taint sources (e.g., untrusted input) as a program executes.

- ➢Taint Introduction
- ➢Taint Propagation
- ➢Taint Checking

```
1. [...]
2. scanf("%d", &x);
3. [...]
4. y = x + k;
5. [...]
6. x = 0;
7. [...]
8. while (i < y)
```

```
1. if (x > 0)
2.     y = 1;
3. else
4.     y = 0;
```

Background

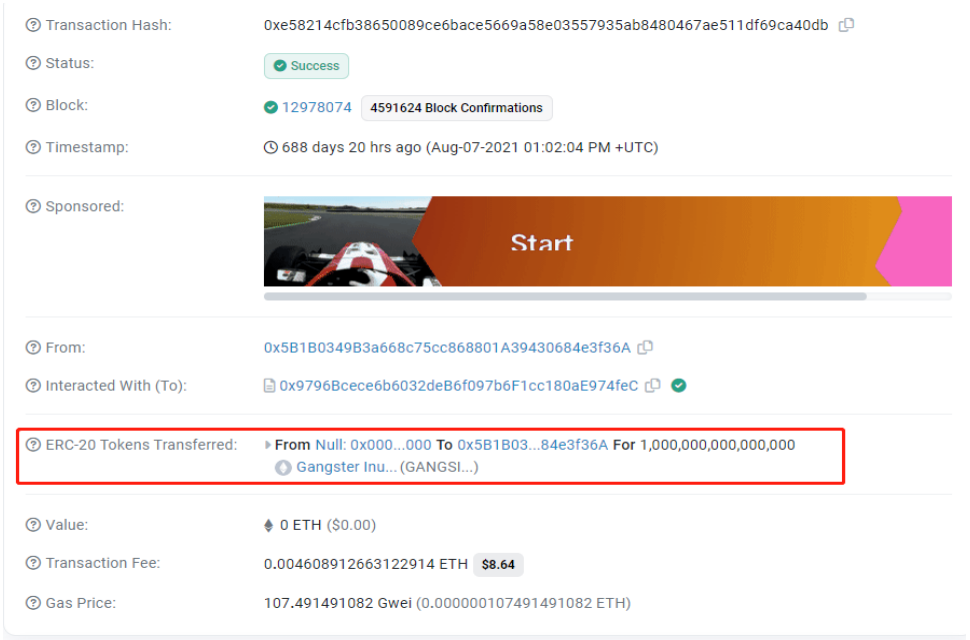
Imitation attack

Naïve string-replace imitation method

```
contract CustomizedLiquidationContract {
  function printMoney(...) public payable {
    require(0x53d8...0d81 == msg.sender);
    //liquidation logic omitted
  }
}
```

This algorithm takes as input a victim transaction, and simply replaces the transaction's sender address with an adversarial address in the transaction sender and data fields.

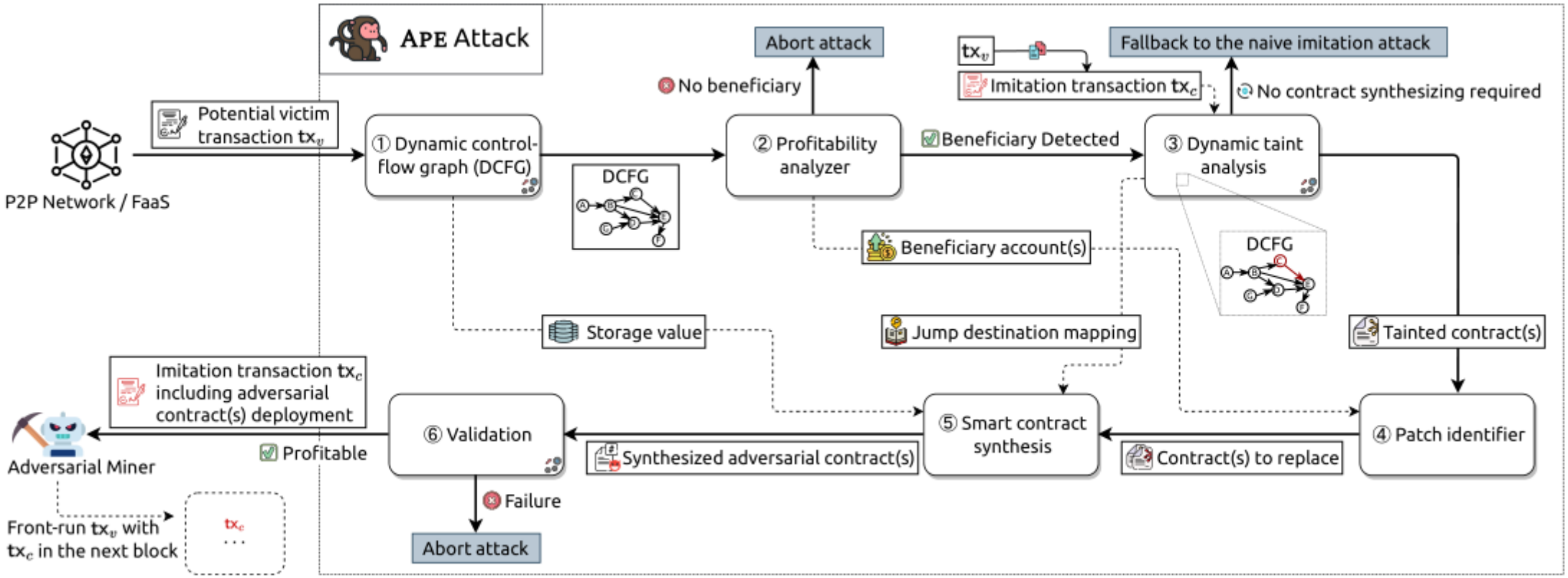
```
contract GANGSINU {
  function increaseAllowance(
    address spender,
    uint256 addedValue
  ) public virtual returns (bool) {
    _approve(
      _msgSender(),
      spender,
      _allowances[_msgSender()][spender] +
          addedValue
    );
    _mint(spender, addedValue);
    return true;
  }
}
```



Problem & Challenge

- ✓short front-running time-window(real-time)
- ✓recursively identify and replace the victim contract
- ✓synthesized contracts are invoked and executed correctly

APE : a generalized imitation tool

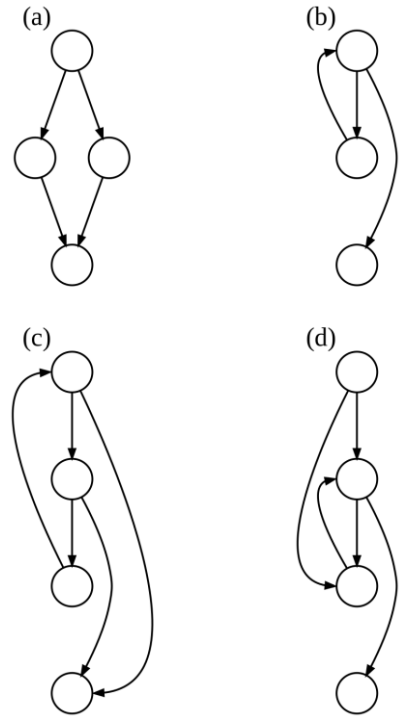


1、Dynamic Control-Flow Graph

~~CFG(static information)~~

DCFG(dynamic information)

execute tx locally



record **condition value of JUMPI**(for 3)

identify contract calls and track the executions of all smart contract

2、Profitability Analyzer

extract the asset transfers from the DCFG constructed through analyzing the EVM logs defined in asset implementation standards

Beneficiary Account

- If the sender is a beneficiary account, other accounts are irrelevant to the profitability analyzer.
- Otherwise, if the sender is not a beneficiary account, the collective profit of other beneficiary accounts, minus the potential loss of the sender account must remain positive.

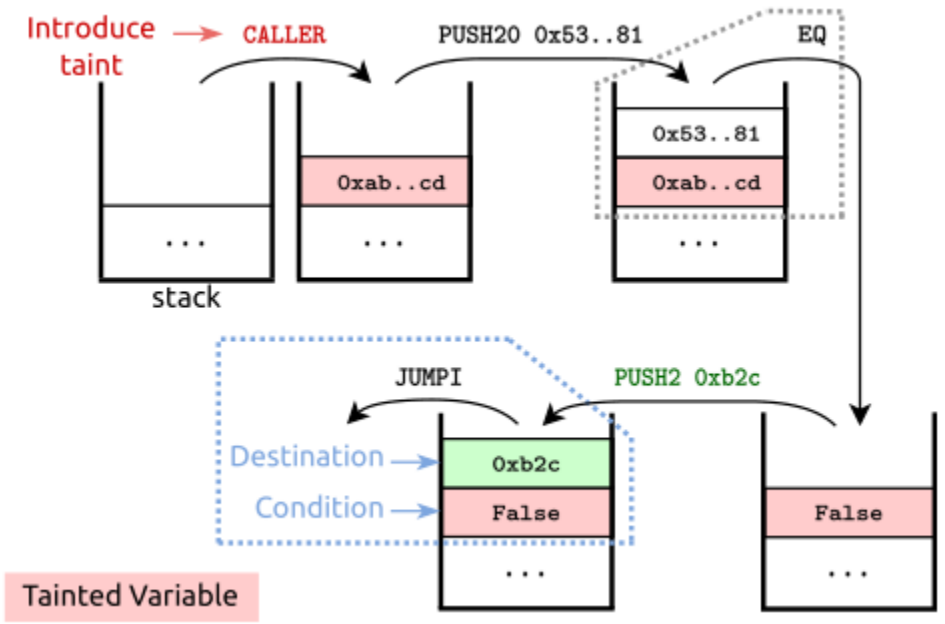
3、Dynamic Taint Analysis

track where and how tx_c's execution fails

```
1  contract CustomizedLiquidationContract {
2    function printMoney(...) public payable {
3      require(0x53d8...0d81 == msg.sender);
4      //liquidation logic omitted
5    }
6  }
```

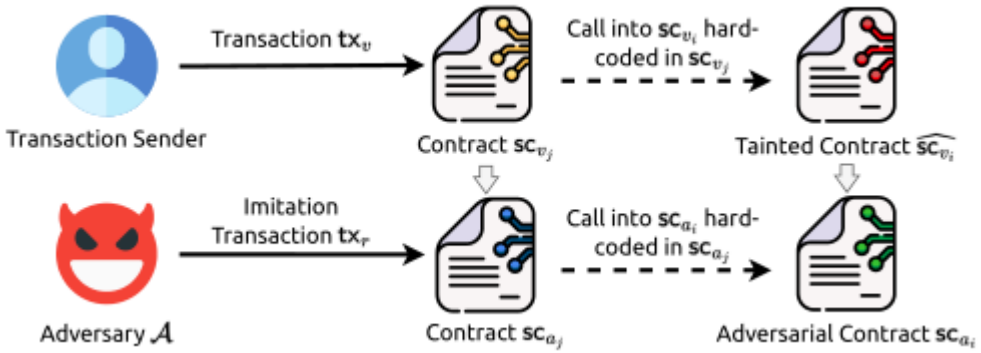
```
    PC: Disassembled Code

0xb0c: JUMPDEST
0xb0d: CALLER
0xb0e: PUSH20     0x53d8...0d81
0xb23: EQ
0xb24: PUSH2      0xb2c
0xb27: JUMPI
0xb28: PUSH1      0x0
0xb2a: DUP1
0xb2b: REVERT
```



4、Patch Identifier

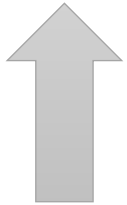
- **Invocation from a transaction**
- **Invocation from a contract**



5、Smart Contract Synthesis

tainted contract: **replaces JUMPI with JUMP**

modify **hard-coded account**(contract)



modify contract bytecode

6、Validation

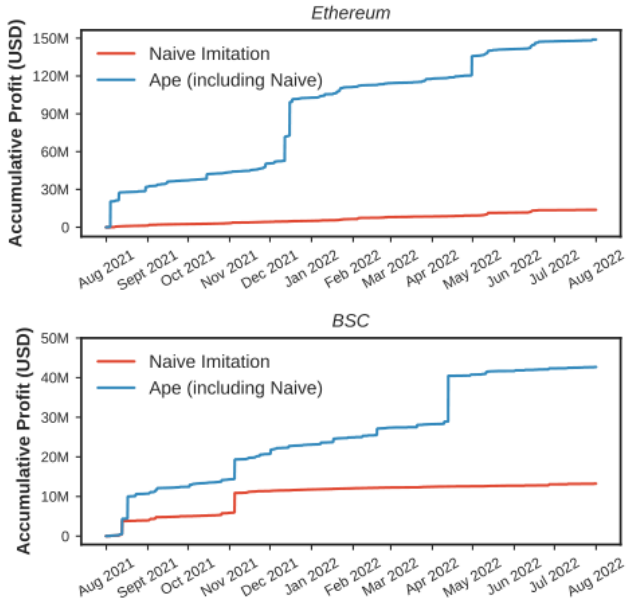
deploys every sc_{ai} and executes tx_c on the latest blockchain state locally

The revenue covers all transaction fees including the smart contract(s) deployment fees.

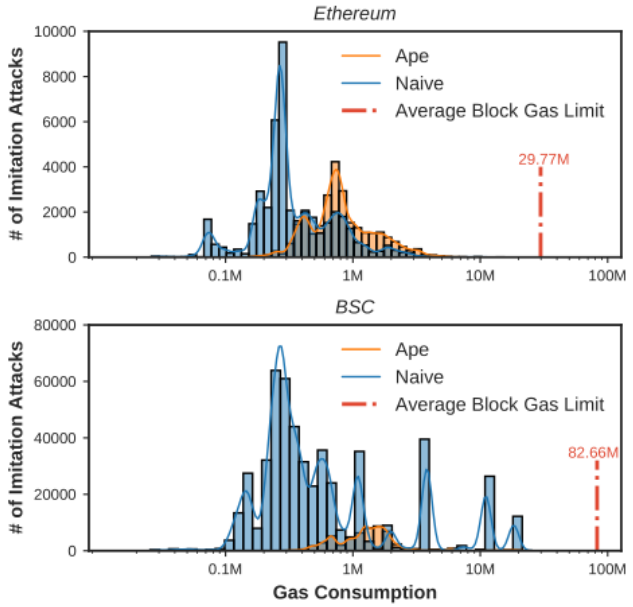
Overall Evaluation Results

Chain	Attack	Transactions	Contracts	Overall Profit (USD)	Average Profit (USD)
Ethereum	Naive	43,979	NA	13.87M	315.48 ± 4.73K
	APE	26,127	665	135.08M	5.17K±227.22K
BSC	Naive	516,128	NA	13.25M	25.67 ± 1.78K
	APE	52,799	1,193	29.45M	557.75 ± 55.88K

Overall attack statistics



Attack profit



Gas consumption

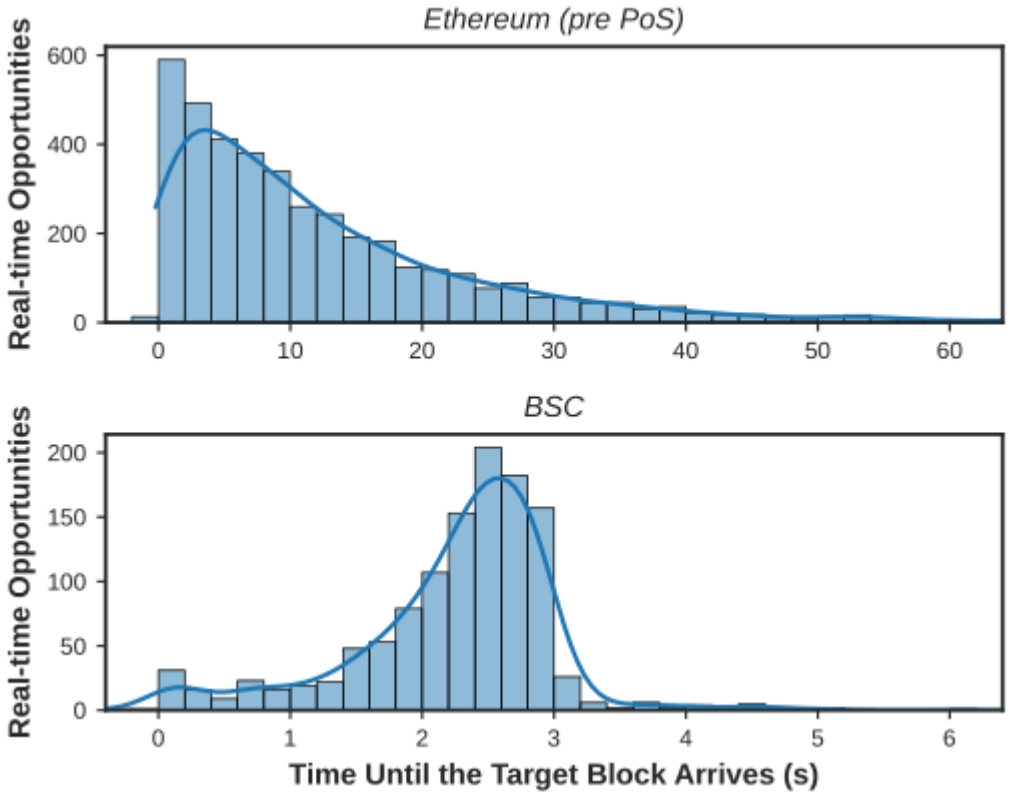
		Mean	Std.	Max	Min
Ethereum	Adversarial Contract Number	1.02	0.15	3	1
	Contract Size Reduction	60.16%	19.19%	98.63%	−295.56%
BSC	Adversarial Contract Number	1.05	0.23	3	1
	Contract Size Reduction	57.59%	18.69%	99.46%	−613.33%

Adversarial contract

APE Real-Time Performance

	Mean (s)	Std. (s)	Max (s)	Min (s)
Step ① DCFG	0.02	0.03	0.36	3×10^{-4}
Step ② Profitability Analyzer	2×10^{-3}	5×10^{-3}	0.10	2×10^{-5}
Step ③ Dynamic Taint Analysis	0.04	0.06	1.39	3×10^{-4}
Step ④ Patch Identifier	2×10^{-5}	5×10^{-5}	2×10^{-3}	1×10^{-6}
Step ⑤ Smart Contract Synthesis	5×10^{-4}	2×10^{-3}	0.09	2×10^{-5}
Step ⑥ Validation	7×10^{-3}	0.02	0.96	2×10^{-4}
Overall time cost of APE	0.07	0.10	1.59	9×10^{-4}
Overall time cost Naive Imitation	0.01	0.01	0.11	2×10^{-4}

Single transaction performance



Mempool performance

APE Countermeasures

- Imitation as a Defence Tool (Whitehat hacking)
- Breaking Atomicity
- Front-running Mitigation
- Code Obfuscation

Conclusion

- generalized blockchain imitation game
- APE: a generalized imitation tool for EVM-based blockchains
- the first to show that dynamic program analysis techniques can realize an imitation attack, posing a substantial threat to blockchain users