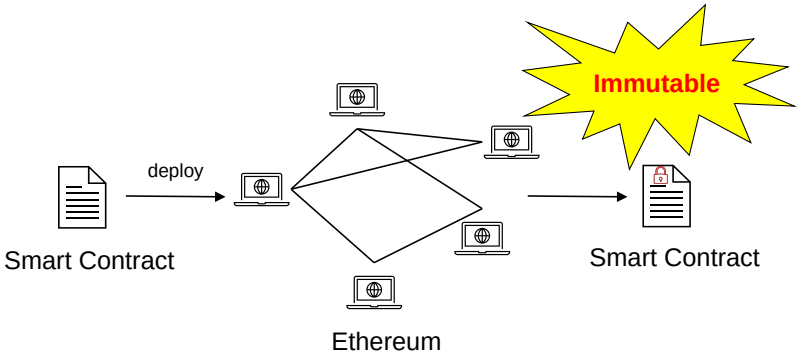
TxSpector: Uncovering Attacks in Ethereum from Transactions

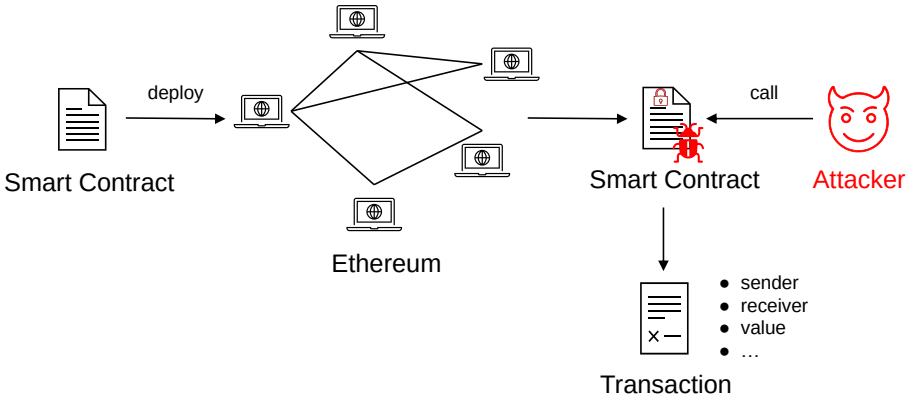
Mengya Zhang, **Xiaokuan Zhang**, Yinqian Zhang, Zhiqiang Lin

USENIX Security 2020

Motivations (1/3) - Deployed Smart Contracts are Immutable



Motivations (2/3) - Attacks can Cause Huge Financial Losses



Motivations (3/3) - Few Works focus on Transactions

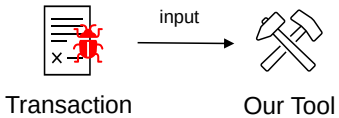
Smart Contracts

1. Oyente [LCO+16]
2. Zeus [KGDS18]
3. Securify [TDDC+18]
4. Vandal [BJK+18]
5. Gigahorse [GBSS19]
6. Maian [NKS+18]
7. Slither [FGG19]
8. Mythril [Con]
9. EthBMC [FAH20]

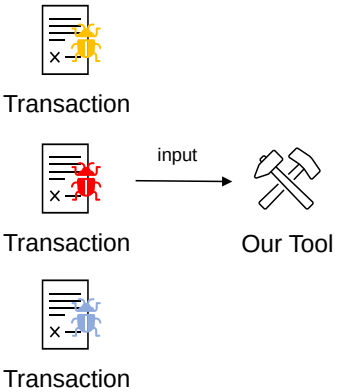
Transactions

1. Sereum [RLKD19]
2. ECFChecker [GAGG+17]

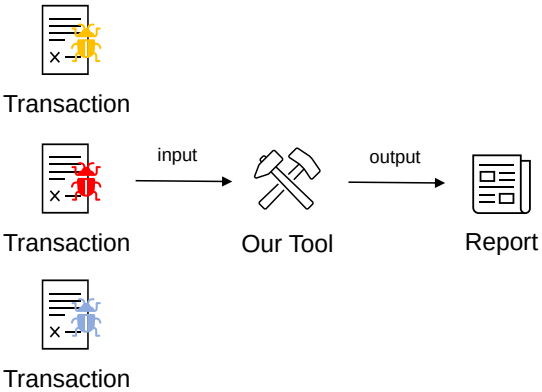
Goals (1/3) - Identify Real World Attacks



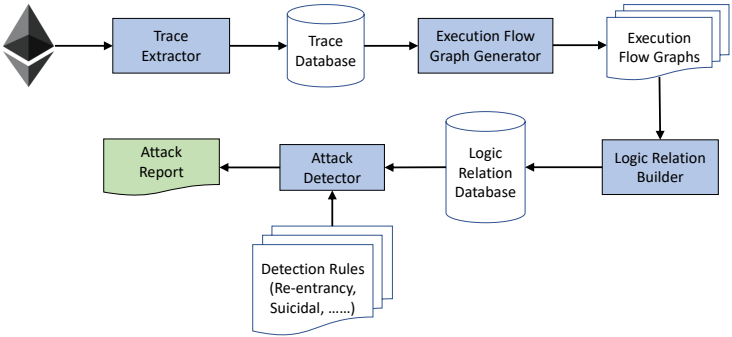
Goals (2/3) - Generic and Logic-driven Framework



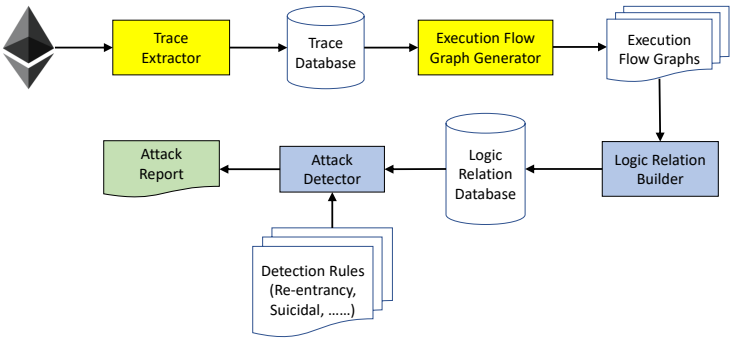
Goals (3/3) - Forensic Analysis of the Attacks



Overview of TxSpector



Detailed Design -Trace Extractor



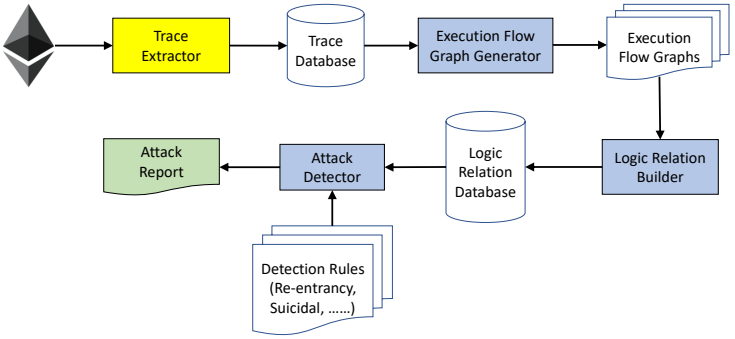
Challenges

- Extract dependencies.

Solutions

- Record bytecode-level traces.

Detailed Design -Trace Extractor



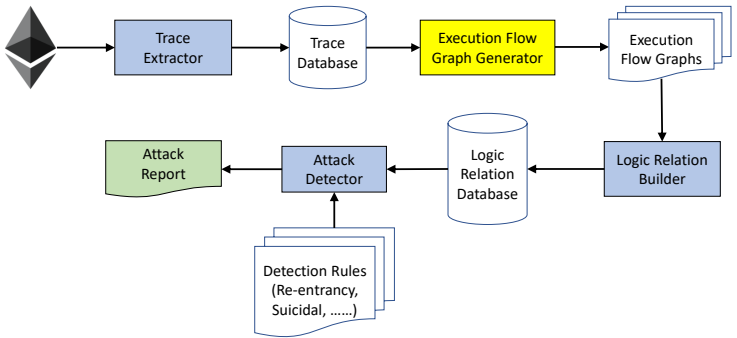
Challenges

- Extract dependencies.

Solutions

- Record bytecode-level traces.

Detailed Design - Execution Flow Graph Generator



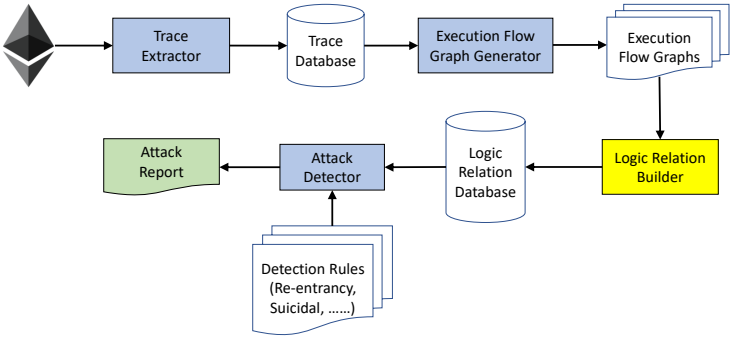
Challenges

- Extract dependencies.

Solutions

- Construct the Execution Flow Graph.

Detailed Design - Logic Relation Builder



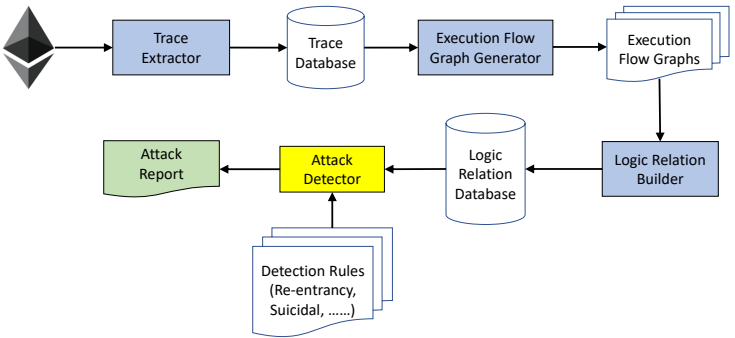
Challenges

- Encode dependencies.

Solutions

- Extract logic relations.

Detailed Design - Attack Detector



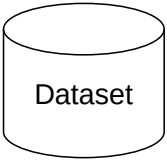
Challenges

- Huge transaction volumes.

Solutions

- Construct once, detect multiple times.

Experiment Setup



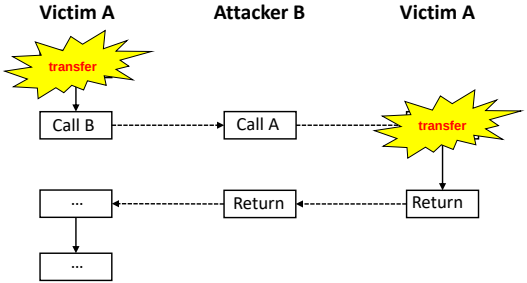
January 2019 – February 2019

9,661,593 transactions

Reentrancy; Unchecked Call; Suicidal

Reentrancy Attacks - An Example

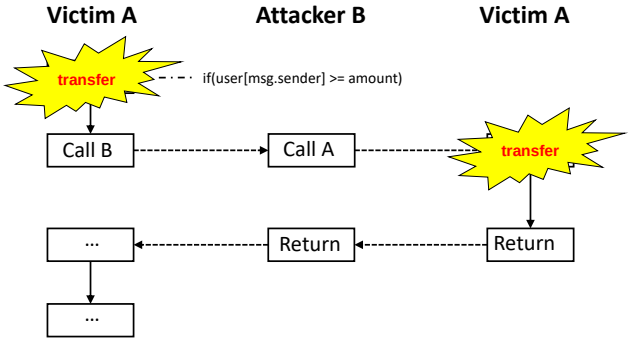
```
1  function withdrawBalance(uint amount) public {
2      if (user[msg.sender] >= amount) {
3          msg.sender.call.value(amount)();
4          user[msg.sender] -= amount;
5      }
6  }
```



Reentrancy Attacks - An Example

inconsistent state

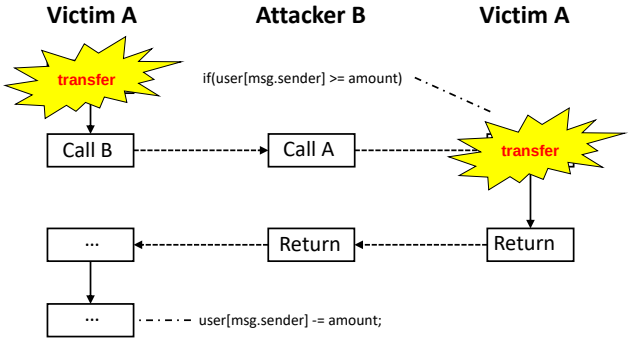
1. **read-if dependency.**
2. read-write dependency.



Reentrancy Attacks - An Example

inconsistent state

1. read-if dependency.
2. **read-write dependency.**



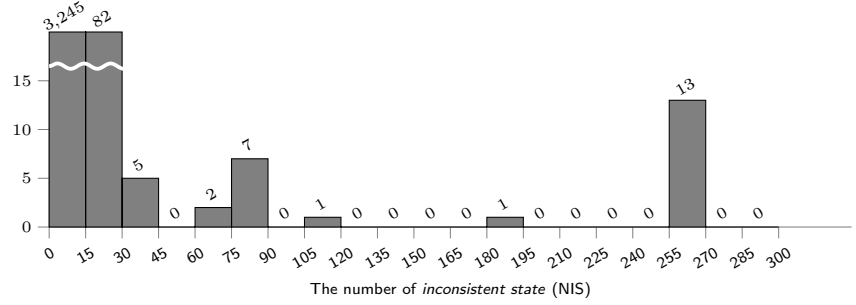
Experiment Results - Reentrancy Attacks

Vulnerability	System	# Total	# Timeout or Error	# Remaining	# Flagged
Reentrancy	TxSpector	9,661,593	336,909 (3.5%)	9,321,684	3,357
	Sereum	9,661,593	N/A	N/A	10,278
	Securify	105,535	7,541	97,994	1196
	Vandal	105,535	1,431	104,104	85,721
	Gigahorse	105,535	N/A	N/A	3,310

Experiment Results - Compare with Other Tools

Vulnerability	System	# Total	# Timeout or Error	# Remaining	# Flagged
Reentrancy	TxSpector	9,661,593	336,909 (3.5%)	9,321,684	3,357
	Sereum	9,661,593	N/A	N/A	10,278
	Securify	105,535	7,541	97,994	1196
	Vandal	105,535	1,431	104,104	85,721
	Gigahorse	105,535	N/A	N/A	3,310

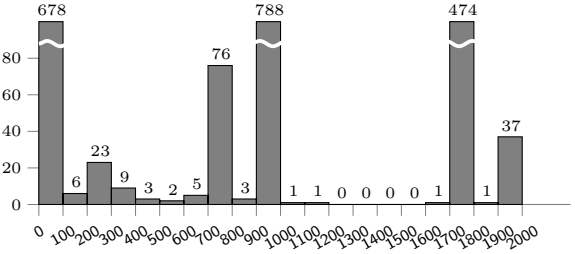
Forensic Analysis - NIS



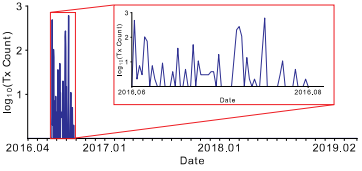
Forensic Analysis - Top 5 Victim Smart Contracts

Address	NIS Count
0xdf18880a02c7f3eb4f40fdf515fce31c1cb7ef66	4,803
0x1806b3527c18fb532c46405f6f014c1f381b499a	3,815
0xd7a14019aeeba25e676a1b596bb19b6f37db74d2	2,839
0x533bafa16aa76218ec4a365ad71bf8816cf21bbb	675
0x431d77f50803d31b090e86740b1d5848af54fad0	582

Forensic Analysis - Case Study: The DAO Smart Contract



NIS in reentrancy transactions on DAO



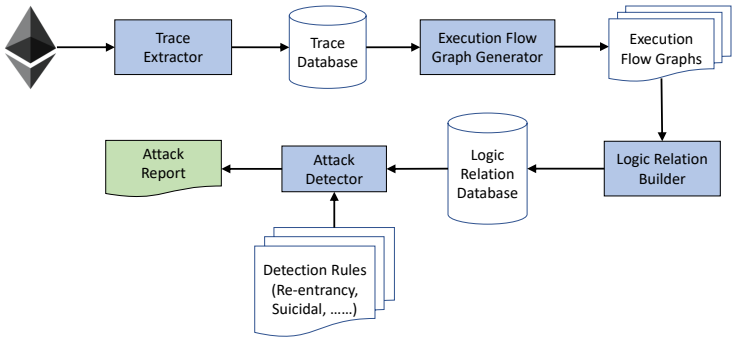
Distribution of reentrancy attacks on DAO.

Related Work

Systems	Tx Order Dependence	State Dependence	Mishandled Exception	Re-entrancy	Restricted Transfer	Failed Send	Unsecured Balance	Misuse-of-origin	Integer Overflow	Suicidal	Denial-of-Service
Oyente [LCO+16]	▲	▲	▲	▲							
Zeus [KGDS18]	▲	▲	▲	▲		▲		▲	▲		
Securify [TDDC+18]	▲		▲	▲	▲						
Vandal [BJK+18]			▲	▲			▲	▲		▲	
Gigahorse [GBSS19]				▲						▲	▲
Maian [NKS+18]					▲					▲	
Slither [FGG19]		▲	▲	▲	▲		▲	▲		▲	
Mythril [Con]	▲	▲	▲	▲		▲	▲	▲	▲		
EthBMC [FAH20]							▲			▲	
Sereum [RLKD19]				★							
ECFChecker [GAGG+17]				★							
TxSpector		★	★	★		★	★	★		★	★

Table: ▲: vulnerabilities in smart contracts; ★: attacks in transactions.

Summary



TxSpector

- Generic, logic-driven, transactions

Applications

- Forensic analysis

Source code is available at: https://github.com/OSUSecLab/TxSpector Email: (zhang.9407, zhang.5840, zhang.834, lin.3021)@osu.edu

References I


Lexi Brent, Anton Jurisevic, Michael Kong, Eric Liu, Francois Gauthier, Vincent Gramoli, Ralph Holz, and Bernhard Scholz, *Vandal: A scalable security analysis framework for smart contracts*, arXiv preprint arXiv:1809.03981 (2018).


ConsenSys, *Mythril classic*, https://github.com/ConsenSys/mythril-classic.


Joel Frank, Cornelius Aschermann, and Thorsten Holz, *ETHBMC: A bounded model checker for smart contracts*, 29th USENIX Security Symposium (USENIX Security 20), USENIX Association, 2020.


Josselin Feist, Gustavo Grieco, and Alex Groce, *Slither: a static analysis framework for smart contracts*, 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), IEEE, 2019.


Shelly Grossman, Ittai Abraham, Guy Golan-Gueta, Yan Michalevsky, Noam Rinetzky, Mooly Sagiv, and Yoni Zohar, *Online detection of effectively callback free objects with applications to smart contracts*, Proceedings of the ACM on Programming Languages (2017).


Neville Grech, Lexi Brent, Bernhard Scholz, and Yannis Smaragdakis, *Gigahorse: Thorough, declarative decompilation of smart contracts*, International Conference on Software Engineering (ICSE), 2019.


Sukrit Kalra, Seep Goel, Mohan Dhawan, and Subodh Sharma, *Zeus: Analyzing safety of smart contracts*, Proceedings of the 25th Annual Network and Distributed System Security Symposium, 2018.


Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor, *Making smart contracts smarter*, Proceedings of the 2016 ACM SIGSAC conference on computer and communications security, ACM, 2016.

References II


Ivica Nikolić, Aashish Kolluri, Ilya Sergey, Prateek Saxena, and Aquinas Hobor, *Finding the greedy, prodigal, and suicidal contracts at scale*, Proceedings of the 34th Annual Computer Security Applications Conference, ACM, 2018.


Michael Rodler, Wenting Li, Ghassan Karame, and Lucas Davi, *Sereum: Protecting existing smart contracts against re-entrancy attacks*, Proceedings of the 26th Network and Distributed System Security Symposium, 2019.


Petar Tsankov, Andrei Dan, Dana Drachsler-Cohen, Arthur Gervais, Florian Buenzli, and Martin Vechev, *Securify: Practical security analysis of smart contracts*, Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2018.