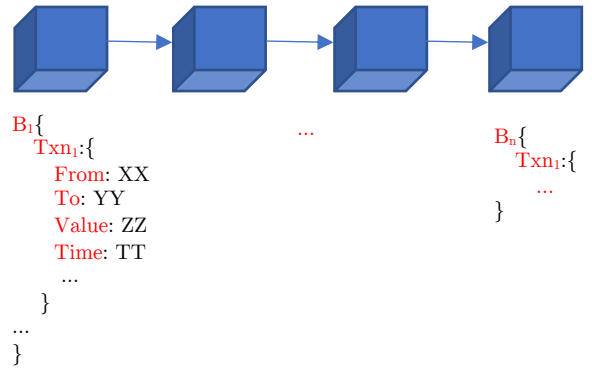
Vulnerability and Transaction Based Detection of Malicious Smart Contracts

CSS 2021
13th International Symposium on Cyberspace Safety and Security
Copenhagen, Denmark (Online)
9-11 November 2021

Authors – Rachit Agarwal,
Tanmay Thapliyal,
Sandeep K. Shukla

Blockchain



- Digital Ledger of Transactions which is:
  - Distributed and duplicated across different computer systems on the network
  - Cryptographically secure – tend to be pseudo anonymous
  - Immutable
- Types of Blockchains
  - Permissionless
    - Anyone can use the blockchain
  - Permissioned
    - Permission is required to use the blockchain
  - Other Types
    - Consortium
    - Hybrid based

Motivation

- More than 10 billion in USD have been received by malicious entities from 2017 – 2020.
- A malicious Smart Contract (SC) can also create many SCs.
- Can such entities be detected using machine learning algorithms?

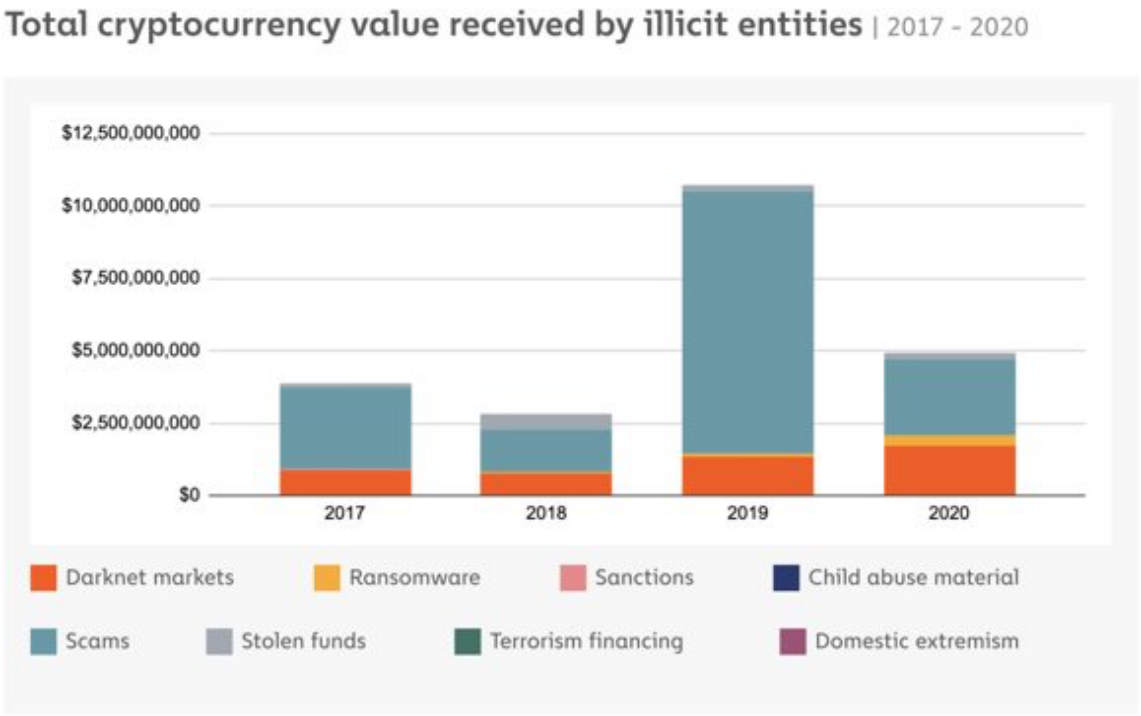


Image source: https://go.chainalysis.com/2021-Crypto-Crime-Report.html

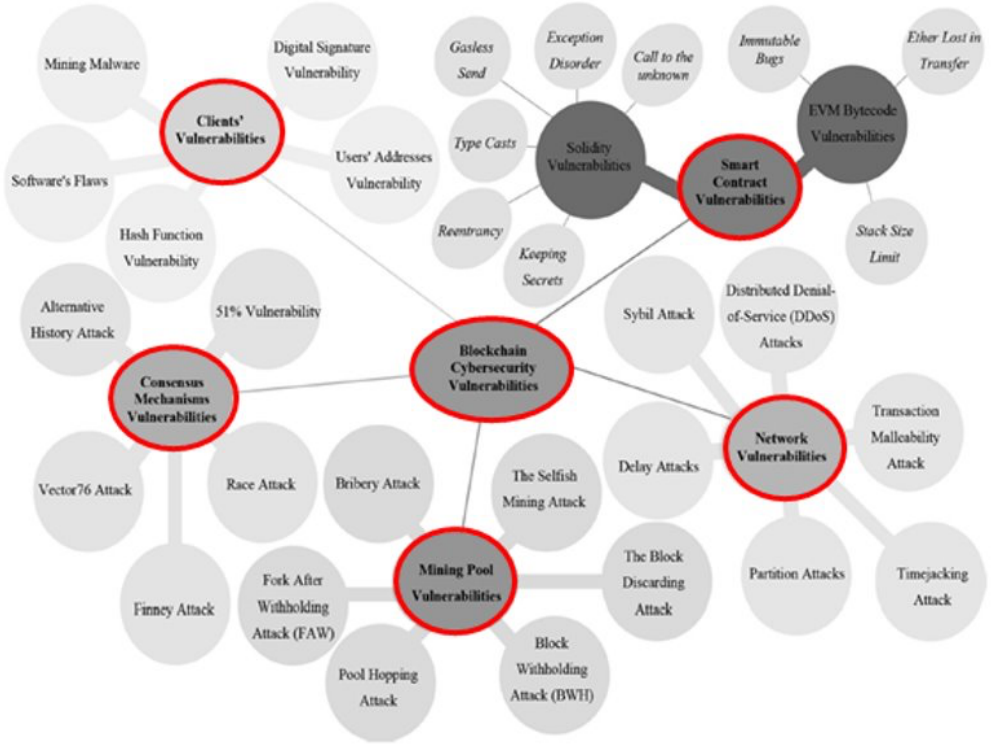


Image source: Blockchain for Cybersecurity and Privacy: Architectures, Challenges, and Applications.

Smart Contracts (SC)

- SCs are code that execute when certain conditions are met.
- SCs can transact with other SCs through internal transactions, which are not recorded on the blockchain.
- SCs can also create other SCs.
- Most SCs written in Solidity (Turing Complete).
- SC vulnerabilities can be classified into: 10 groups using DASP[1] taxonomy and 34^{+} classes using SC Weakness Classification (SWC)[2] and corresponding Common Weakness Enumeration (CWE)[3]

Severity	Vulnerability	DASP-10	SWC	CWE	Severity	Vulnerability	DASP-10	SWC	CWE
H	Arbitrary-send†	Acc. Control	124	123	H	Uninitialized state†	Unknown	109	824
H	Ether send ‡		105	284	H	Uninitialized storage†		109	824
H	Unprotected self destruct† ‡ ◊		106	284	H	Shadowing state†		119	710
H	Delegate call†‡		112	829	H	Locked Ether◃†		-	-
H	tx-origin◃ ‡ †		115	477	M	Uninitialized local†		109	824
H	Integer Overflow‡◊ • ‡	Arith.	101	682	M	Constant function†		-	-
H	Integer Underflow‡◊•		101	682	M	Shadowing abstract†		119	710
M	Signedness bugs•		101	682	M	ERC20 returns false◃		135	1164
M	Truncation bugs•		101	682	M	Incorrect Blockhash◃		104	252
M	Callstack bug•◊	DoS	113	703	M	Balance Equality†◃		132	697
M	Overpowered role◃		-	-	L	Usage of Assembly†◃		-	695
M	Gas Limit in Loops◃		128	400	L	Pragmas version◃		102	937
M	Transfer in Loop◃†		113	703	L	Should not be view◃		-	-
L	Array Length Manipulation ◃		128	400	L	Bad Visibility◃		108	710
L	Multiple Calls‡		113	703	L	Shadowing-builtin†		119	710
H	Reentrancy-eth†#	Reentrancy	107	841	L	Shadowing-local†		119	710
M	Message call to ext. contract‡		107	841	L	Hardcoded address◃		-	547
M	Call without data◃		107	841	L	Deprec. Constructions◃		111	477
M	Reentrancy-no-eth†#		107	841	L	Extra gas in loops◃		128	400
L	Reentrancy-benign†#		107	841	L	Redun. fallback reject◃		135	1164
L	State change after ext. call‡		107	841	L	Revert require◃		123	573
H	Unchecked call return value◃‡	U.L.C	104	252	L	Exception State‡		110	670
M	Unused return†		135	1164	M	TOD‡		114	362
L	Send◃		104	252	M	Timestamp mani.◃ † ‡ • ◊		116	829

† = Slither, ‡ = Mythril, ◃ = SmartCheck, ◊ = Oyente, • = Osiris, ^{severity} H = high, M = medium, L = low, - inferred by us and not directly present in SWC and CWE, -: not present in the vocabulary, $^{Acc.}$ Access, Arith.: Arithmetic, U.L.C.: Unchecked Lowlevel Calls

[1] https://dasp.co
[2] https://swcregistry.io
[3] https://cwe.mitre.org

Overview

- Our aim in this work is to answer the following questions:
  - Is there a correlation between a particular malicious activity and a vulnerability in the SC, and if so, does the severity of such vulnerability correspond to its exploitability in committing malicious activities.
  - Does the severity score of a vulnerability accounts for an important feature towards detecting malicious accounts.
  - Do SCs not marked malicious behave maliciously across different temporal granularities when severity scores are considered as a feature.

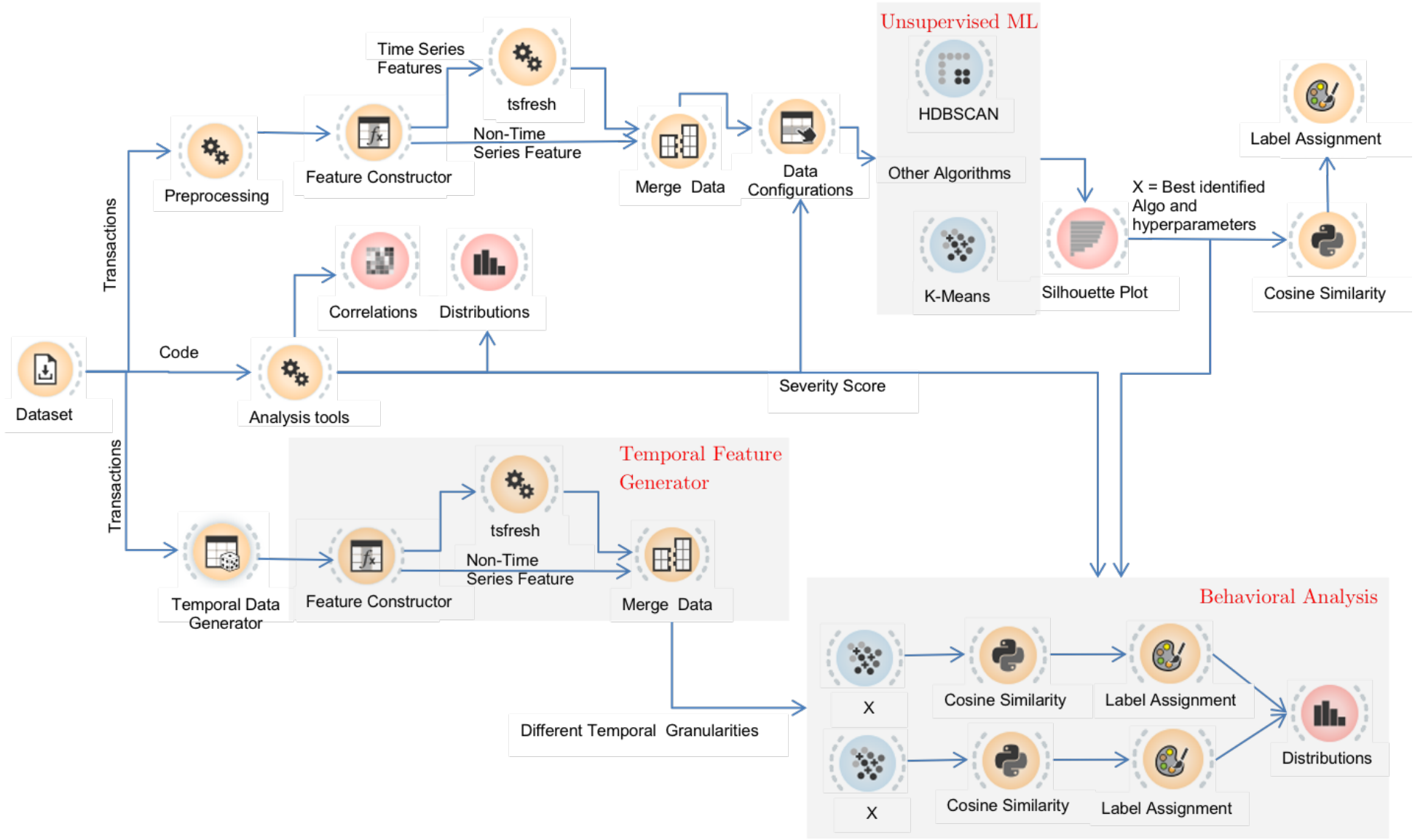
Related Works

- **Transactions specific**
  - State of the art have used both temporal and graph based transaction features to detect malicious accounts in blockchains.
  - Do not take into consideration the internal transactions of a smart contract[4,5]

- **Vulnerability specific**
  - Different works develop and compare different tools that can detect vulnerabilities in SCs.
  - Study the effect of exploitation of different SC vulnerabilities.
  - Do not consider transaction behavior of SCs[6,7]

[4] R. Agarwal, et al, Detecting malicious accounts in permissionless blockchains using temporal graph properties. Applied Network Science 6(9), 1–30 (02 2021)
[5] S. Farrugia, et al, Detection of illicit accounts over the Ethereum blockchain. Expert Systems with Applications 150, 113318 (07 2020)
[6] T. Durieux, et al, Empirical Review of Automated Analysis Tools on 47,587 Ethereum Smart Contracts. In: 42nd IEEE Intern. Conf. on Software Engineering. pp. 530–541. Seoul, South Korea (06 2020)
[7] M. Angelo, et al, A survey of tools for analyzing Ethereum smart contracts. International Conference on Decentralized Applications and Infrastructures. pp. 69–78. IEEE, Newark, CA (08 2019)

Methodology



Features Used

- Temporal Features[4]
  - **Burst**: non uniformity in a time-series.
    - Balance Burst, GasPrice Burst, Degree Burst
  - **Attractiveness**: Change in the stability of the neighborhood of a smart contract.
- Severity score based on the vulnerabilities present in the SCs.
  - An SC can have multiple vulnerabilities with varying severity.
  - The severity score is defined by :

$$Ss^i = \frac{\sum_{\forall j \in V^i} S_j}{|V^i|}$$

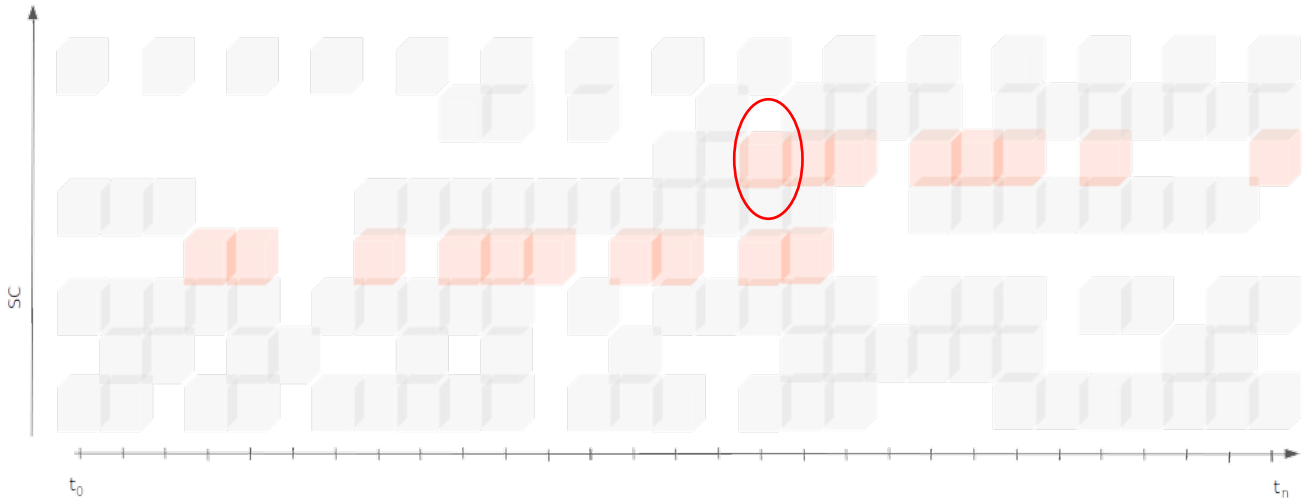
Ss^i: Severity Score for each SC
V^i: Vulnerability in an SC
S_j: Severity associated with the vulnerability (high=3, medium=2, low=1)

[4] R. Agarwal, et al, Detecting malicious accounts in permissionless blockchains using temporal graph properties. Applied Network Science 6(9), 1–30 (02 2021)

Behavioral Analysis

- Different temporal granularities of data-frames are created to study the temporal behavior of accounts.
- Motivation behind this is that in each granularity an account may change its behavior.
  - A benign account may act malicious on some days.

Representation of SCs w.r.t time



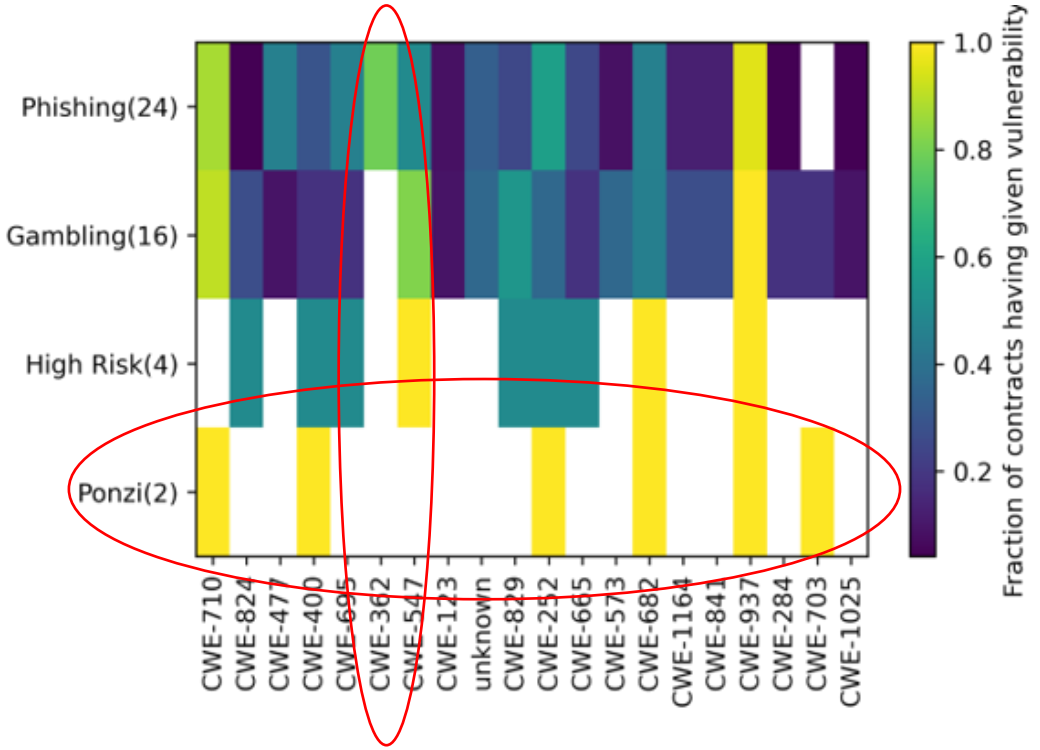
Dataset

- 47398 benign SCs with unique source codes[5].
- 46 malicious SCs with unique source code are identified from SCs with source code amongst 1.4 million smart contracts whose source code are verified by Etherscan[8].
- Tools used for detection of vulnerabilities in SCs: Mythril, SmartCheck, Slither, Oyente, Osiris.
- Segment/Dataframe distribution:
  - 1791 in 1-day granularity
  - 598 in 3-day granularity
  - 60 in 1-month granularity

[5]T. Durieux, et al, Empirical Review of Automated Analysis Tools on 47,587 Ethereum Smart Contracts. In: 42nd IEEE Intern. Conf. on Software Engineering. pp. 530–541. Seoul, South Korea (06 2020)
[8]https://etherscan.io

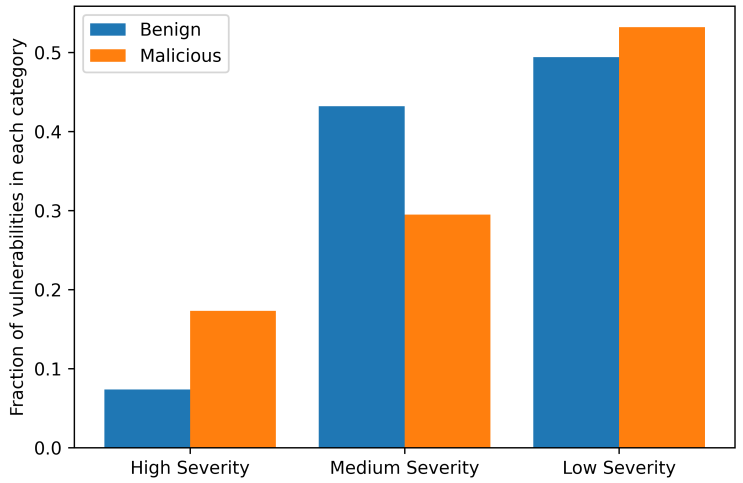
Results

- Transaction Order Dependence (TOD) vulnerability is mostly present in SCs involved in Phishing. While Phishing SCs do not have DOS.
- Gambling SCs do not have TOD.
- SCs related to Ponzi schemes do not have Reentrancy vulnerability.



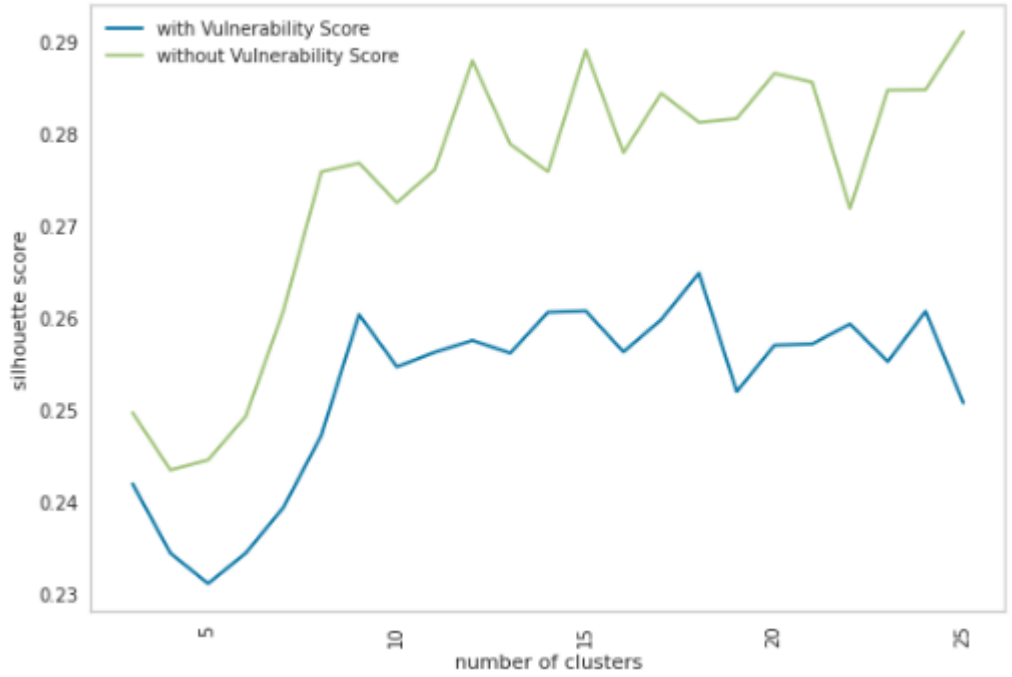
Results (cont...)

- We also observe that high severity vulnerabilities are present in benign SC, but their fraction is less than that in malicious SCs.
- Since the difference between the fraction for malicious and benign class SCs for each severity category is very small, we cannot say that vulnerability implies exploitability.



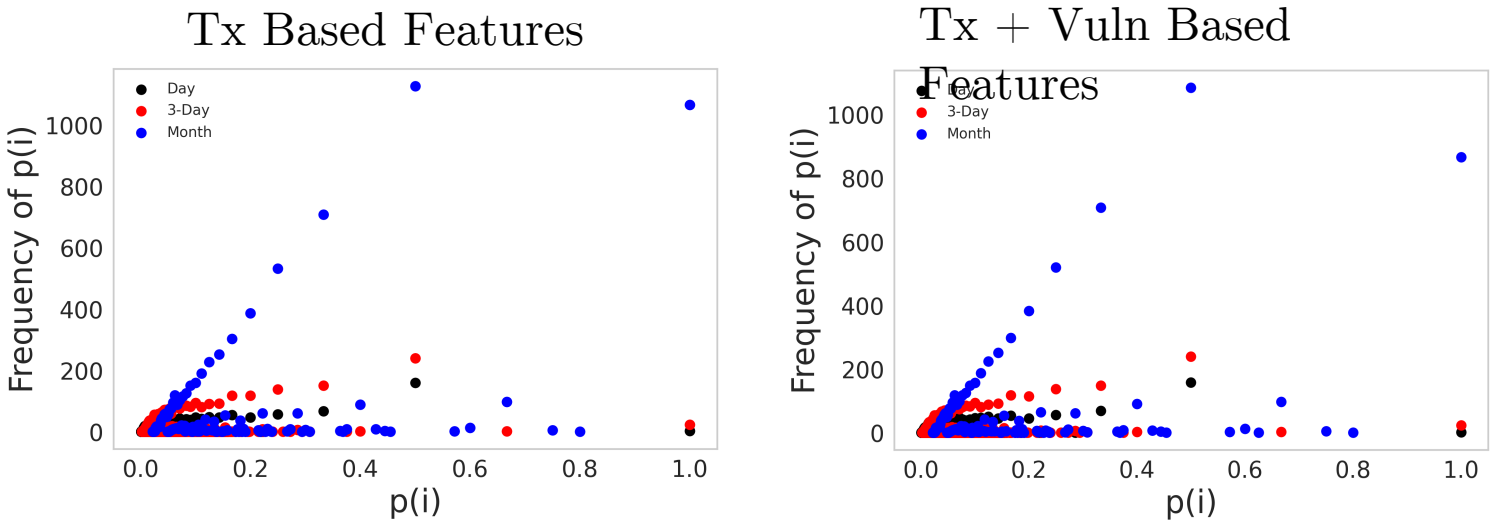
Results (cont...)

- K-Means acquires best silhouette score in our dataset.
- The silhouette score decreases when vulnerability score is introduced as a feature in our dataset.
- We infer that when such severity scores are considered a feature vector, the data is either more uniformly distributed or more densely distributed, causing overlapping clusters. The clusters thus formed are indistinguishable from each other, which in turn reduces the silhouette score.



Results (cont...)

- We compute probability of an SC being malicious in different temporal granularities.
- 866, 24 and 2 SCs identified as suspects when only transaction-based features considered in 1-month,3-day and 1-day granularity.
- 1066, 24, and 4 SCs identified as suspects when both transaction and vulnerability-based features considered in 1-month,3-day, and 1-day granularity.
- No common suspect SCs between different granularities. Thus, behavior of SCs changes across different temporal granularities.



Conclusion and Future Work

- Illicit activities exploit different vulnerabilities
- No significant correlation between vulnerability and exploitability
  - Benign and Malicious SC's both have vulnerabilities
- Transaction behavior of SCs changes across different temporal granularities
- Homogenizing SC vulnerability vocabulary

- Recommendation
  - Scrutiny of Smart Contracts a must.

- Future work
  - Discover robust ways to compute severity scores.

Thank You

Questions

tanmayt@iitk.ac.in
@ThapliyalTanmay
rachitag@iitk.ac.in
@ragarwa2