Phantom Blocks

Attacks leaving trails behind

Presented by : Sai Krishna
CS21B1020
IIIT Raichur
Guided by : Dr. Pradhan

Introduction

- Ethereum blockchain sufferred from multiple attacks and different anomalies.
- Aims to identify a few attacks over the network.
- Show how these attacks effected the network
- How methods has been developed to nulify these attacks on Ethereum
- Helps in future prediction of attack behorehand to avoid financial loss, congestion etc.

Results of previous work

- Mini 1 with Bhavya

We have done data analytics over ethereum network, to understand how the network operates and how real world instances effected the ethereum.

- Mini 2 with Bhavya

Also i have done, node classification over ethereum previously because understanding what kind of nodes have which kind of features, plays a key role for identifying anomalous nodes.

Problem statement

- How to identify attack situations in Ethereum network?
- What kind of vulnerabilities play crucial role in this identification?
- What data analytics or Machine learning techniques are required? Is machine learning required?
- What thresholds need to be taken for vulnerability identification?

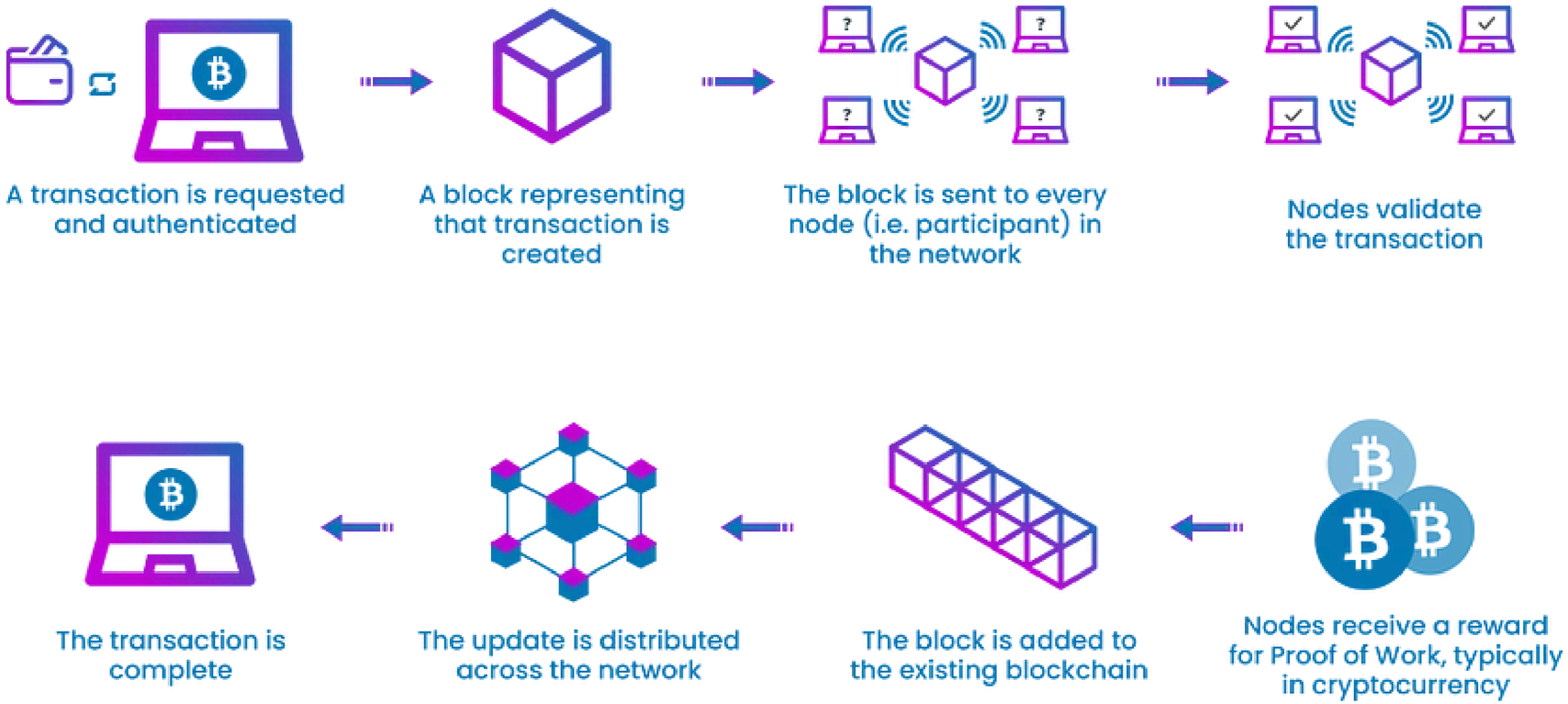
Literature Review

- Lawal, Y. L. (2020). Anomaly Detection in Ethereum Transactions Using Network Science Analytics (Master's thesis, University of Cincinnati).

- Chen, Huashan, Marcus Pendleton, Laurent Njilla, and Shouhuai Xu. "A survey on ethereum systems security: Vulnerabilities, attacks, and defenses." ACM Computing Surveys (CSUR) 53, no. 3 (2020): 1-43.

- Chen, T., Li, X., Wang, Y., Chen, J., Li, Z., Luo, X., Au, M.H. and Zhang, X., 2017. An adaptive gas cost mechanism for ethereum to defend against under-priced dos attacks. In Information Security Practice and Experience: 13th International Conference, ISPEC 2017, Melbourne, VIC, Australia, December 13–15, 2017, Proceedings 13 (pp. 3-24). Springer International Publishing.

Ethereum Transactions



Reference: https://coinsbench.com/a-comprehensive-guide-to-blockchain-technology-working-and-security-3647d15678ea

Ethereum

- Ethereum can do many tasks other than financial transactions
- Dapps, Games, etc many run on this network. But how??
- Two types of accounts
  -Smart contracts
  -Externally owned accounts
- Smart contract is self executing code when particular agreements are met

Why we do this?

- This particular application handles billions of dollars in it
- Detecting attacks earlier can help us to prevent loss of funds
- Also prevents slowing of network and avoids congestion
- Helps in taking precautions so that the network wont face such attacks again

DAO hack-2016

- token holders could then vote on which projects to fund, meaning the organization was supposed to be fully governed by its community, without central authority.
- Has 14% of ether($150 million) at that time.
- Attacker stole $50 million
- Error in code, drains fund before updating balance.
- Ether price dropped
- Network broke into two parts ETH(change accepted) and ETC(didnt accept change)

ICO Mania-2017

- Raising funds by offering tokens and took funds in ether
- Some attracted users by giveaways and airdropping etc
- Account creations are cheap
- Too many accounts created to claim free tokens
- Condition to revert the transaction if fixed fund is not received for fixed time by user
- Also noticed too many revert transactions

GovernMental contract attack-2016

- Paid old investors from new investors
- Pyramid mechanisms
- Caused looping of many transactions or chain transactions
- Too much gas is used increasing gas price
- Resulted in too many out of gas transactions
- Funds got locked in the application
- Finally resulted in DoS

Cryptokitties-2017

- Game for trading, holding, breeding cats
- Had referral bonuses
- So many fake accounts were created which spammed the network for account creation
- Also this game required to many transactions to happen at each and every development in game process.
- Slowed network by spam account creation,Congested network due to these transactions
- Increased the gas prices

Architecture

- Identified various patterns and after effects of each attack
- extracted the neccessary transaction data with such patterns
- plotted them for data analytics
- Tried to apply machine learning to identify the anomalies
- Tried various thresholds for finding the optimal one

Methodology

Data preprocessing

- Extracted the data from Xblock.pro\Xblock-Eth website from 2015-08-07 to 2019-01-02
- Data is of csv format as shown

blockNum	timestamp	transactio	from	to	toCreate	fromIsCon	toIsContra	value	gasLimit	gasPrice	gasUsed	callingFunc	isError	eip2718ty	baseFeePe	maxFeePe	maxPriorityFeePerGas
46147	1.44E+09	0x5c504ec	0xa1e438(	0x5df9b87	None	0	0	31337	21000	5E+13	21000	0x	None	None	None	None	None
46169	1.44E+09	0x19f1df2	0xbd08e0(	0x5c12a8e	None	0	0	1.99E+19	21000	9.1E+11	21000	0x	None	None	None	None	None
46170	1.44E+09	0x9e6e19(	0x63ac545	0xc93f225	None	0	0	6E+20	21000	5E+11	21000	0x	None	None	None	None	None
46194	1.44E+09	0xcb9378(	0x037dd05	0x7e7ec15	None	0	0	1E+20	21000	1E+12	21000	0x	None	None	None	None	None
46205	1.44E+09	0x570ce19	0x3f2f381	0x4bd5f0e	None	0	0	8.04E+20	21000	5E+11	21000	0x	None	None	None	None	None
46214	1.44E+09	0xe17d4d(	0xa1e438(	0xc9d4035	None	0	0	31337	21750	5E+13	21748	0x74796d3	None	None	None	None	None
46217	1.44E+09	0x2ec3825	0xc8ebccc	0xc8ebccc	None	0	0	0	21000	6.53E+10	21000	0x	None	None	None	None	None
46219	1.44E+09	0xe891897	0xa1e438(	0x5df9b87	None	0	0	31337	21800	5E+13	21748	0x74796d3	None	None	None	None	None
46220	1.44E+09	0x35d4f3d	0xf0cf0af5	0xb608771	None	0	0	1E+20	21000	6.42E+10	21000	0x	None	None	None	None	None
46230	1.44E+09	0x417387(	0x1c68a66	0xc8ebccc	None	0	0	5E+19	21000	7.13E+10	21000	0x	None	None	None	None	None
46235	1.44E+09	0x80f3170	0xfd2605a	0x073f70b	None	0	0	1E+16	21000	7.06E+10	21000	0x	None	None	None	None	None
46237	1.44E+09	0x3a1be27	0xbbed465	0xbf8d8b4	None	0	0	4.41E+12	21000	7.05E+10	21000	0x	None	None	None	None	None
46239	1.44E+09	0xc0c1c72	0x8ce4945	0x15e34a8	None	0	0	1E+20	21000	1E+12	21000	0x	None	None	None	None	None
46240	1.44E+09	0x04ff148	0x136d4b6	0xc8ebccc	None	0	0	1E+21	21000	8.14E+10	21000	0x	None	None	None	None	None
46242	1.44E+09	0x8e2ba7(	0x4d92795	0x99c2361	None	0	0	1E+18	21000	7.47E+10	21000	0x	None	None	None	None	None

- Converted timestamp and splitted data into day wise files

2016-02-14	23-04-2024 11:32	Microsoft Excel Co...	2,547 KB
2016-02-15	23-04-2024 11:32	Microsoft Excel Co...	3,461 KB
2016-02-16	23-04-2024 11:32	Microsoft Excel Co...	3,543 KB
2016-02-17	23-04-2024 11:32	Microsoft Excel Co...	3,528 KB
2016-02-18	23-04-2024 11:32	Microsoft Excel Co...	3,966 KB

Data preprocessing

- Removed unneccessary columns to reduce the size of data

blockNumber	timestamp	transactionHash	from	to	toCreate	fromIsContract	toIsContract	value	gasLimit	gasPrice	gasUsed	callingFunction	isError
46147	1438918233	0x5c504ed432cb51138bcf09aa5e8a410dd4a1e204ef84bfed1be16dfba1b22060	1	2	0	0	31337	21000	50000000000000	21000	0x		
46169	1438918613	0x19f1df2c7ee6b464720ad28e903aeda1a5ad8780afc22f0b960827bd4fcf656d	3	4	0	0	1990000000000000000	21000	909808707606	21000	0x		
46170	1438918630	0x9e6e19637bb625a8ff3d052b7c2fe57dc78c55a15d258d77c43d5a9c160b0384	5	6	0	0	5999895000000000000	21000	500000000000	21000	0x		
46194	1438918983	0xcb9378977089c773c074045b20ede2cdcc3a6ff562f4e64b51b20c5205234525	7	8	0	0	10000000000000000000	21000	1000000000000	21000	0x		
46205	1438919175	0x570ce19176bd0002b04a9179309129bbdaf0c4252ffeb76aedbb038cdf662850	9	10	0	0	8039895000000000000	21000	500000000000	21000	0x		
46214	1438919394	0xe17d4d0c4596ea7d5166ad5da600a6fdc49e26e0680135a2f7300eedfd0d8314	1	11	0	0	31337	21750	50000000000000	21748	0x74796d34		
46217	1438919451	0x2ec382949ba0b22443aa4cb38267b1fb5e68e188109ac11f7a82f67571a0adf3	12	12	0	0	0	21000	65334370444	21000	0x		
46219	1438919461	0xe891897177614c91284b6929dc2ada5a87705c4729bda2d5eeae47ea2ca9d175	1	2	0	0	31337	21800	50000000000000	21748	0x74796d34		
46220	1438919491	0x35d4f3dae18d72a0d4caf02359ca1844687ff879a2655bdec5c33c9b0f65f795	13	14	0	0	10000000000000000000	21000	64178193561	21000	0x		
46230	1438919571	0x41738785c4330ce9531aed26b21b9cfba6f27b9183d11355f4d9522211c2a44e	15	12	0	0	5000000000000000000	21000	71288549894	21000	0x		
46235	1438919655	0x80f31704782a53514ab0693499f78922169885a16fd4821d42a4a820d2452799	16	17	0	0	10000000000000000	21000	70563255618	21000	0x		

- Now consider the from and to columns in the dataset. Convert hash into numerical values and store them in a text file. Same hash has same number everywhere.

from	to
0x39fa8c5f2793459d6622857e7d9fbb4bd91766d3	0xc083e9947cf02b8ffc7d3090ae9aea72df98fd47
0x32be343b94f860124dc4fee278fdcbd38c102d88	0xdf190dc7190dfba737d7777a163445b7fff16133
0x2a65aca4d5fc5b5c859090a6c34d164135398226	0x819f4b08e6d3baa33ba63f660baed65d2a6eb64c
0x2910543af39aba0cd09dbb2d50200b3e800a63d2	0x9e486ad335492959c38a0740cb66c55ad30bb4f0
0x10d5bff7879b7eb5192b3374338bb834981910a8	0xc6c764fc6c1e1211d2b4a06ef2170f660a4512fa
0x2a65aca4d5fc5b5c859090a6c34d164135398226	0x53e0551a1e31a40855bc8e086eb8db803a625bbf
0x2a65aca4d5fc5b5c859090a6c34d164135398226	0x51033f1a1a59cb6a1bf6ca2087a53bd202ac1c83
0x120a270bbc009644e35f0bb6ab13f95b8199c4ad	0x3dc12a32a5abf477e2ec91f6218d0a96150fef99
0x2a65aca4d5fc5b5c859090a6c34d164135398226	0xf4f2c15602b084cae84ea603f75527de19705aa1

-->

1	2
3	4
3	145
3	66
3	66
3	146

Data preprocessing

- Mapping calling function hashes also in similar manner and storing them in another file

- and mapped isError column also like if it is normal transaction -0

  out of gas -2

  revert -1

blockNumber	timestamp	transactionHash	from	to	toCreate	fromIsContract	toIsContract	value	gasLimit	gasPrice	gasUsed	callingFunction	isError
46147	1438918233	0x5c504ed432cb51138bcf09aa5e8a410dd4a1e204ef84bfed1be16dfba1b22060	1	2	0	0	31337	21000	50000000000000	21000	1	0	
46169	1438918613	0x19f1df2c7ee6b464720ad28e903aeda1a5ad8780afc22f0b960827bd4fcf656d	3	4	0	0	1990000000000000000	21000	909808707606	21000	1	0	
46170	1438918630	0x9e6e19637bb625a8ff3d052b7c2fe57dc78c55a15d258d77c43d5a9c160b0384	5	6	0	0	5999895000000000000	21000	500000000000	21000	1	0	
46194	1438918983	0xcb9378977089c773c074045b20ede2cdcc3a6ff562f4e64b51b20c5205234525	7	8	0	0	1000000000000000000	21000	1000000000000	21000	1	0	
46205	1438919175	0x570ce19176bd0002b04a9179309129bbdaf0c4252ffeb76aedbb038cdf662850	9	10	0	0	8039895000000000000	21000	500000000000	21000	1	0	
46214	1438919394	0xe17d4d0c4596ea7d5166ad5da600a6fdc49e26e0680135a2f7300eedfd0d8314	1	11	0	0	31337	21750	50000000000000	21748	2	0	

Feature extraction

- total transactions present in this period are 4,40,96,502
- Found too many new account created which have no use until now after creation as a vulnerability situation(vulnerability-1)
- Considered only to accounts which were present in the to position once and never present again in any position
- 29,45,819 such accounts were found
- Extracted Gaslimit, Gasused, from node, block number, timestamp for those account transactions.

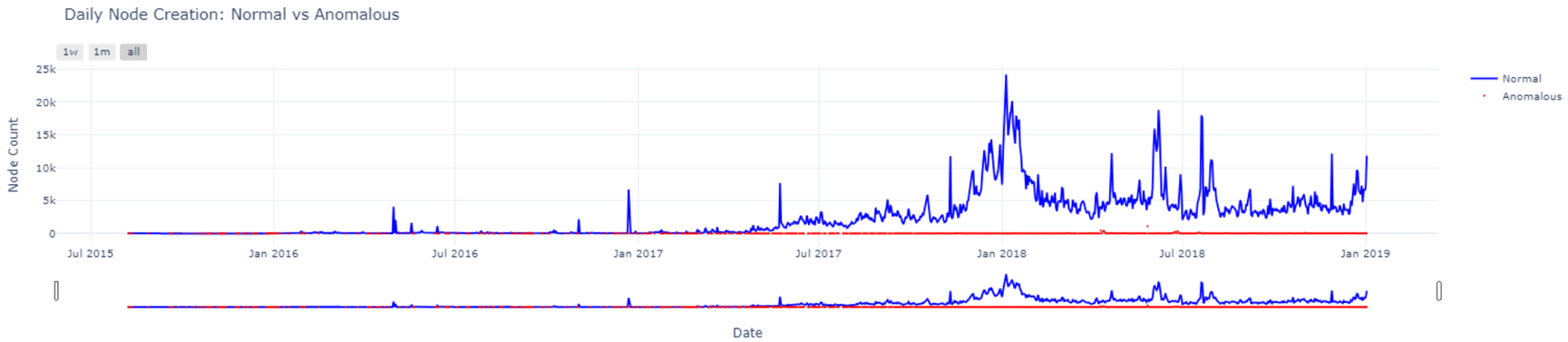
Feature extraction

- Extracted out of gas transactions over the period as count of out of gas transactions vulnerability-2
- 22,92,143 such transactions are present
- along with out of gas transactions, i have extracted the block number and timestamp of these transactions
- Extracted revert transaction over the period as i have considered revert transactions as vulnerability-3
- 64,13,343 revert transactions are present
- also block number and timestamp of the transactions are extracted

Results

- Did K-means over the extracted data of vulnerability 1 with gas used, gaslimit, block number,timestamp as features to it with number of clusters 2

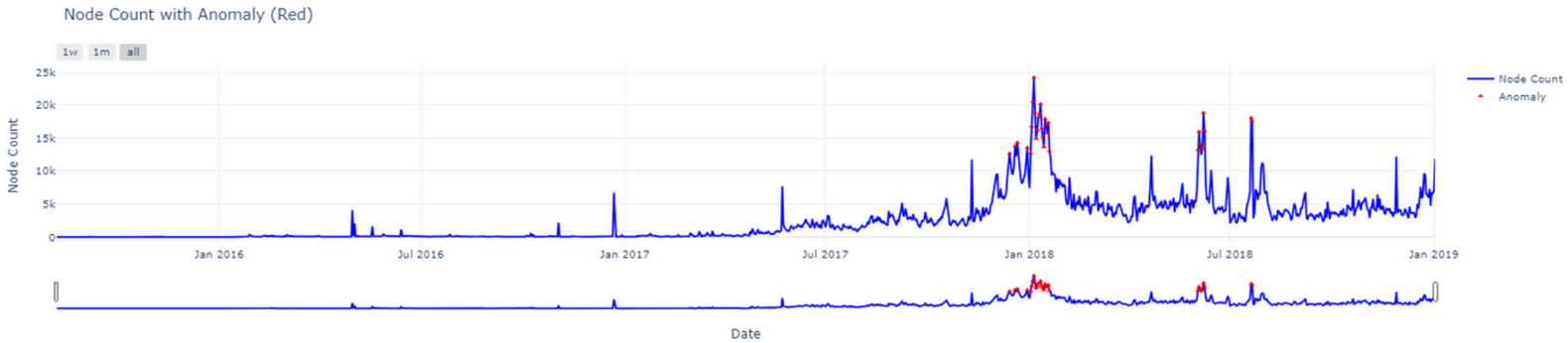
- Plotted the anomaly cluster transactions count per day vs date to find any pattern in it, so that it can be used for attack identification



- But the minor cluster didnt have any pattern or trend to identify the transaction

Results

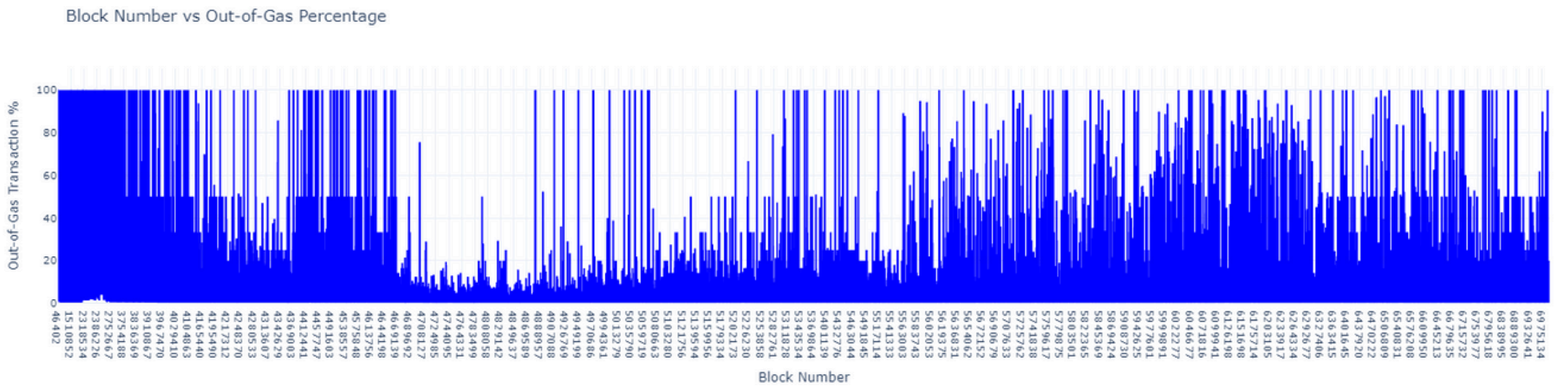
- So, now i have classified count of such accounts per day by converting the timestamp only to date format
- plotted it count vs date
- Applied z-score for anomalies identification and took z-score>3 as anomalies



- This identified anomalies during the attack days, so i considered this as a valid threshold.

Results

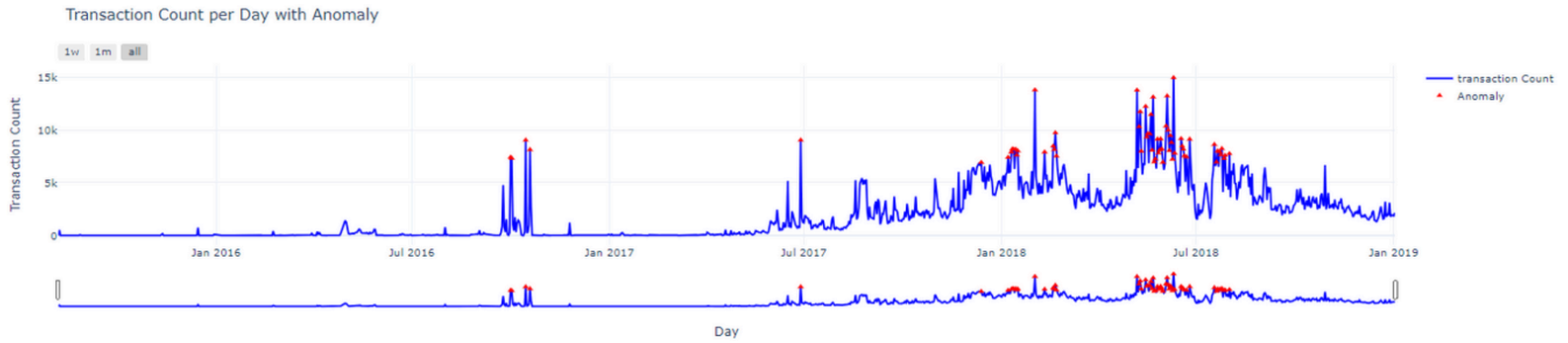
- For identifying vulnerability-2, i have taken the count of such transactions per block and divided with the total number of transactions in that block



- In an attempt to find pattern, i plotted percentage vs block number but found most of them are filled with similar transactions

Results

- Now plotted day vs out of gas transaction count and applied z-score>2

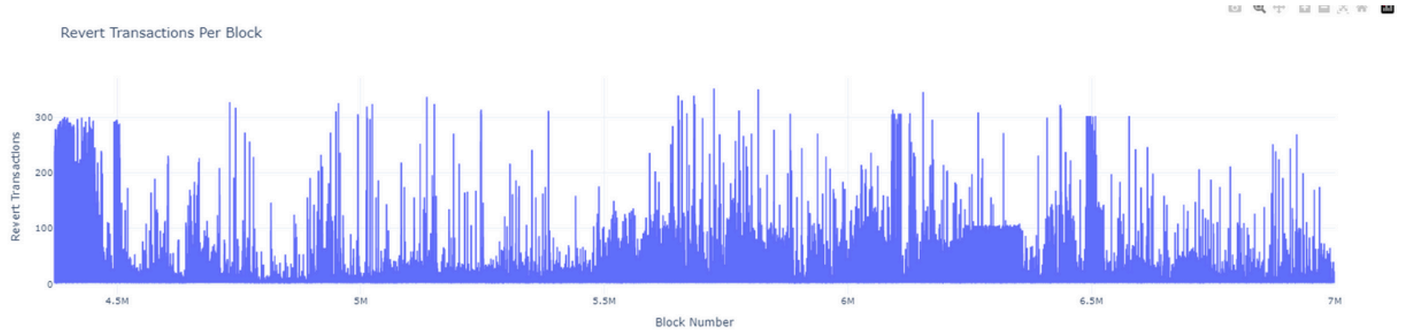


- Here, the identified anomalies are during attack day which helped me in validating the threshold.

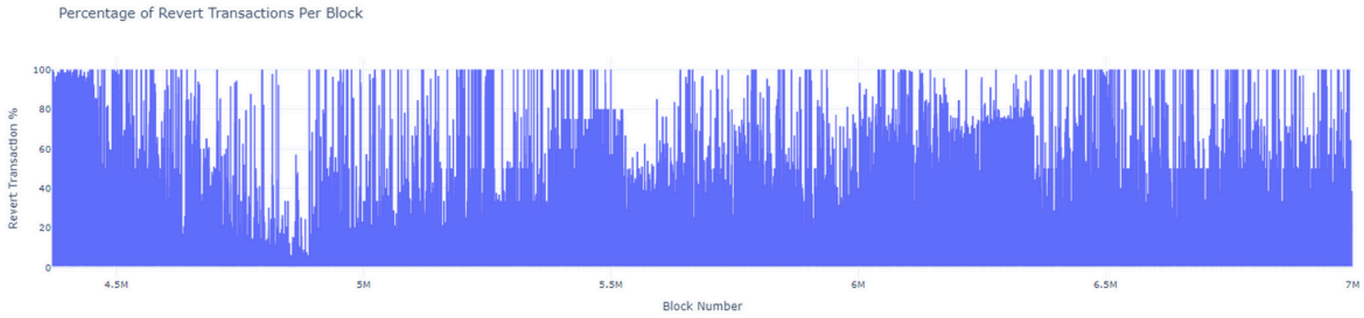
- For this, Z-score>3 failed to identify the pattern

Results

- For identifying the vulnerabilty-3, i plotted revert transaction count vs block number

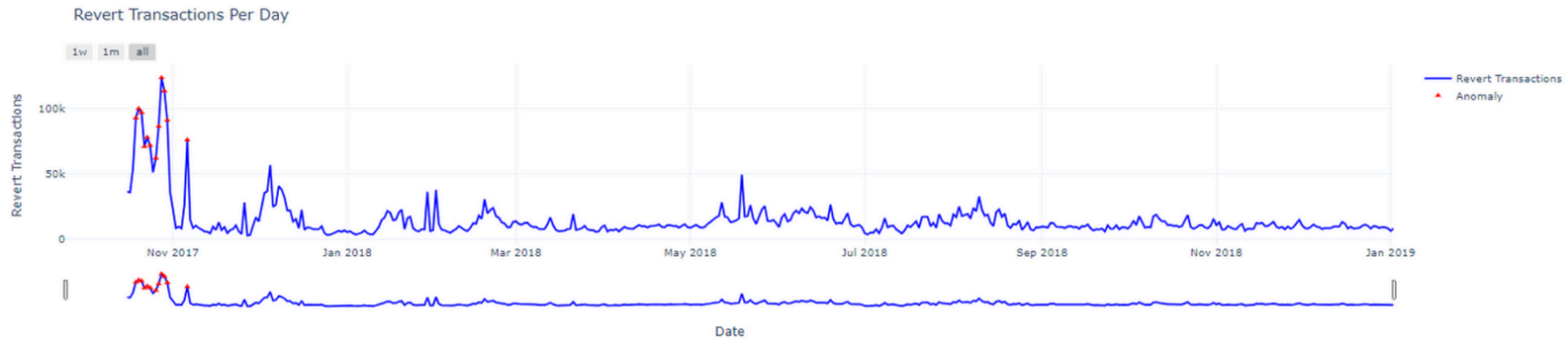


- For identifying vulnerability-3, i have taken the count of such transactions per block and divided with the total number of transactions in that block



Results

- But the percentage didnt helped us in identifying the pattern and as a result i plotted the count vs day and took z-score >3



- Here, the revert transaction count matched with DAO hack days aftereffect. So, the threshold is valid
- Now, you can assume these 3 feilds can help you in attack identification

Challenges faced

- No proper resources for identification of attack
- complex structure of ethereum and huge data format
- choosing methods and thresholds for anomaly detection
- Large time for computation
- Computer with high GPU power is mostly prefered or else results in memorry error.

Conclusion

- Identified various attacks like DAO, governmental contract attack etc
- attack pattern detection helps in improving blockchain security, performance, and trust.
- understand how attackers manipulate the network during high-activity periods.
- Integrate features from this study into static and dynamic analysis tools to warn developers about gas loop risks, logical flaws, or exploit-prone patterns before deployment.

Thank You