Bigger than We Thought: the Upbit Hack Gang

anonymous authors

anonymous institutes

Abstract. The prosperous development of Ethereum has bred many illegal activities by malefactors, such as Ponzi schemes, theft of funds from exchanges, and attacks on service providers. Aiming to expedite the realization of their gains, criminals will launder illicit money, making it as difficult as possible for security companies or agencies to recover those illicit funds. In this paper, we focus on a typical security event on Upbit exchange and explore the scale of the gang behind the security event. Specifically, We construct a rough suspicious money laundering transaction network by crawling downstream transactions of 815 accounts marked as Upbit hacks. Then, in order to refine a more accurate gang of Upbit hacks, we design a suspiciousness indicator for money laundering and modify an existing general risk assessment framework based on propagation models to assess the money laundering risk of accounts. Based on the risks, we acquire an accurate gang for Upbit hacks. In the end, we find that the size of the Upbit hack gang is much bigger than we thought. We also present several interesting analyses of the Upbit hack gang.

Keywords: Ethereum · Upbit hack · money laundering · suspiciousness indicator.

1 Introduction

Ethereum is the largest blockchain platform that supports smart contracts [1], and the popularity of Ethereum has sparked a significant increase in investment activity. Numerous investors register the account without cost on Ethereum to make transactions with Ether [2] (the native cryptocurrency on Ethereum) and tokens [3] (other cryptocurrencies on Ethereum). To date, the market capitalization of Ethereum has reached about 214 billion USD[1].

However, due to the inadequate law enforcement on Ethereum, criminals are drawn to participating in illicit activities. Once the addresses of criminals acquire ill-gotten gains, the criminals will make every effort to launder money to cash out. The goal of money laundering (ML for short) in cryptocurrency is to move funds to addresses where the original criminal source can not be detected, and eventually to some service providers (such as exchanges, mixers, and loaning pools) that allow cryptocurrency to be exchanged for cash. More than

[1] https://coinmarketcap.com/currencies/Ethereum/

66.2 billion USD worth of cryptocurrency from 2017 to 2022 has been laundered by criminals [4].

ML is rampant, but the accounts used by illegal organizations for ML are generally concealed and unknown. For example, disreputable Upbit hacks stole 342,000 Ether from the Upbit exchange on November 27, 2019. There are only 815 addresses labeled as *Upbit hack* by a prominent blockchain explorer XBlock[2]. By examining the transaction records of those labeled accounts, it can be seen that those illegal accounts have transferred ill-gotten money to other accounts, finally laundering the illegal money to the service providers. It can be inferred that more than just those 815 labeled accounts are involved in the illegal organization's financial activities.

In this paper, we construct a rough suspicious ML transaction network by crawling downstream transactions of 815 accounts marked as Upbit hacks. Then, to refine a gang of Upbit hacks, we design an ML suspiciousness indicator for accounts and modify an existing general risk assessment framework based on propagation models to calculate the ML risks of accounts. Based on ML risks, we filter out a gang for Upbit hacks. In the end, we find that the size of the Upbit hack gang is much bigger than we thought. We also present several interesting findings of ML in the Upbit hack gang.

The multi-faceted contributions of the paper are summarized as follows:

- **A new ML suspiciousness indicator for accounts:** We have designed an indicator called *ML_Suspic* that measures the degree of suspiciousness of an account's involvement in ML by combining the characteristics of account ML on Ethereum.
- **A complete ML dataset of Upbit hacks:** We crawled and constructed a rough suspicious ML transaction network and refine the gang of Upbit hacks. Researchers can conduct more in-depth research based on this dataset. Our dataset is open-sourced at https://www.dropbox.com/scl/fo/8j1otjrnsa5b019pbp2xy/h?dl=0&rlkey=jivsf9ymaenzfj47to17fi5dd
- **Insights on ML in Upbit hack gang:** We analyze the risk distribution of the Upbit hack gang and share some analyses of layering accounts, service providers, and a local ML process in the gang.

2 Rough ML Network Construction

To obtain the entire ML gang of the Upbit hacks, the first step is to construct a rough ML network. We crawl transaction records of the relevant accounts by a heuristic method. Then, we construct those accounts and transactions into a rough ML network.

2.1 Crawling Tool and Event

The Upbit hack event that occurred on November 27, 2019, serves as a prominent example of ML on Ethereum, in which about 342,000 Ether was stolen. The

[2] https://www.xblock.pro/cloud

perpetrators used a large number of accounts to transfer the funds to various service providers in layers, eventually being able to withdraw the funds. We have collected the dataset of Upbit hacks by utilizing a popular blockchain explorer XBlock and an open-source crawler toolkit called BlockchainSpider [5]. Within XBlock's Label Word Cloud, there are 815 accounts that have been labeled as related to the Upbit hacks incident.

2.2 Account and Transaction Data Crawling

To obtain the ML dataset, we utilized a heuristic method approach. ML generally consists of two types of transactions: layering transactions and integration transactions [6]. Layering transactions refer to the process of transferring dirty money to other accounts in order to obscure the funds' source through a layer-by-layer method. Integration transactions refer to the process of transferring illicit money to service providers, such as an exchange or a mixing service, intending to withdraw legal money.

To be specific, first, we utilized BlockchainSpider to crawl transaction records of 815 accounts that have been labeled as related to the Upbit hacks incident. We filtered out transactions that occurred before the incident and were traded with tokens. Then, we classified transaction records into layering transactions and integration transactions, based on whether the receiving accounts were categorized as service providers, such as wallets and loaning pools. On the one hand, integration transactions, which indicate successful ML, are identified when the receiving accounts are classified as service providers. On the other hand, transactions involving non-service providers are considered layering transactions and suggest that the ML process is still ongoing. We continued to collect transferring-out transactions from non-service providers until we arrived at integration transactions.

2.3 Network Construction

After obtaining accounts and transactions, we modeled them as a multi-directed weighted transaction network $G_{rough}= (N, E)$, where N and E represent the set of all accounts and transactions respectively. Each transaction is represented as a four-tuple (u, v, a, t), which means that at timestamp t, account u transfers Ether in the amount of a to account v, where $u, v \in N$. In the G_{rough}, there are more than two hundred thousand accounts and eight hundred thousand transactions, which is a huge gang. Among the transactions, there are five hundred thousand layering transactions and three hundred thousand integration transactions.

3 ML Network Refinement

In order to obtain a more accurate ML gang, we refine the rough network by quantifying the ML risk of accounts and removing accounts whose ML risk was below a threshold and related transactions from G_{rough}.

Quantifying the ML risks of accounts mainly involves two parts. The first part is to design an ML suspiciousness indicator for accounts based on ML characteristics [7,8]. The second part is to combine the suspiciousness indicator with a risk assessment framework [9] to calculate the ML risks of accounts in the rough ML network.

3.1 Design an ML Suspiciousness Indicator

ML accounts usually have two characteristics: (i) zero-out accounts, and (ii) fast-in and fast-out accounts:

Characteristic 1. ML accounts are zero-out middle accounts. Most received money from ML accounts will be transferred out, which leads to the balance of ML accounts being close to zero [7].
Characteristic 2. ML accounts are fast-in and fast-out accounts. Most of those accounts transfer money in and out within a short interval [8].

Based on those two characteristics, we designed an ML suspiciousness indicator for accounts to measure the degree of suspiciousness of the accounts' involvement in ML. We measure the ML suspiciousness of accounts by calculating the balance of accounts (which reflects whether accounts are zero-out middle accounts) and the average rate of in-out transaction pairs (which reflects whether accounts are fast-in and fast-out accounts).

- **Balance of an account**

The balance of an account n is defined as

$$B(n) = In(n) - Out(n), (n \in N),$$

where $In(n)$ represents the total amount of Ether received by n and $Out(n)$ is the total amount of Ether moved out from n.

- **Average rate of transaction pairs of an account**

The average rate of transaction pairs of an account n is defined as

$$A(n) = \frac{\sum T(e_{pair}(n))}{N(e_{pair}(n))}, (e_{pair} \in E),$$

where $e_{pair}(n)$ is an in-out transaction pair of n, $T(e_{pair}(n))$ is the interval between incoming and outgoing transactions of an in-out transaction pair of n, and $N(e_{pair}(n))$ is the number of transaction pairs of n. For instance, one Ether is transferred into n, and after one minute n transfers it to another account. Those two transactions are a $e_{pair}(n)$. One minute is a $T(e_{pair}(n))$.

- **ML suspiciousness of an account**

We have calculated $B(n)$ and $A(n)$ to measure whether the account has the characteristics 1. and 2. respectively, but the range of $B(n)$ and $A(n)$ are different. In order to standardize the range of those two indicators, We first make use of a logarithmic scale to transform the values of those two indicators, which helps to narrow the gap between values, and then apply min-max normalization to make those indicators in the range [0, 1].

The Normalized $B(n)$ is represented as

$$Nor\_B(n) = \frac{\log(B(n)+1)}{\log(Max\_B+1)}, (n \in N),$$

where Max_B is the maximum value of $B(n)$ in G_{rough}, the reason why we shift the indicator values to the right by one unit is to make $Nor\_B(n)$ not smaller than zero.

Similar to $Nor\_B(n)$, the Normalized $A(n)$ is represented as

$$Nor\_A(n) = \frac{\log(A(n)+1)}{\log(Max\_A+1)}, (n \in N).$$

Then, we use the arithmetic mean of $Nor\_B(n)$ and $Nor\_A(n)$ to calculate the ML suspiciousness. The ML suspiciousness of an account n is defined as

$$ML\_Suspic(n) = \frac{1}{2}(Nor\_B(n) + Nor\_A(n)), (n \in N), \tag{1}$$

$ML\_Suspic(n)$ ranges from 0 (low suspicious) to 1 (high suspicious).

3.2 Calculate ML Risks of Accounts

To calculate ML risks of accounts, we combined the ML suspiciousness with the Riskprop model [9]. By incorporating the ML suspiciousness into the Riskprop model and modifying the model, we acquire an ML risk for each account. The pseudo-code of ML risk calculation is described in Algorithm 1.

The ML risk ranges from 0 (very low risk) to 10 (very high risk). We have divided the risks into four levels based on values of risks, which are low risk ([0,4)), moderate risk ([4,6)), moderate-high risk ([6,8)), and high risk ([8,10]). We believe that accounts with a risk level below 4 and an average transaction amount of accounts less than 0.01 Ether are innocent accounts, and we remove those innocent accounts and corresponding transactions from G_{rough}.

4 Results and Analysis

After calculating ML risks of accounts in G_{rough}, we remove innocent accounts and corresponding transactions from G_{rough}. Finally, we acquire a $G_{refined}$ which is consist of 7,914 accounts and 109,572 transactions, which is a huge gang.

As shown in Fig. 1, we have analyzed an ML risk distribution of accounts in $G_{refined}$ and obtained the following observations and analyses: 1) In the entire refined ML network $G_{refined}$, there are 6,459 high-risk accounts far exceeding the 815 accounts labeled as Upbit hacks, which indicates that the $G_{refined}$ has added more suspicious accounts related to ML on the basis of the original 815 Upbit hacks. 2) There are a few moderate-risk accounts in $G_{refined}$, those moderate-risk

Algorithm 1 Accounts ML Risk Calculation Algorithm

```
Input: Multi-directed Weighted Transaction Network $G_{rough} = (N, E)$
Output: ML Risks of accounts
Initialize $Trust^0 = 0.5$, $Reliable^0 = 0.7, t = 0, \Delta = 1$
Calculate ML suspiciousness of accounts using Equation(1)
Calculate score of accounts using an equation in [9]
while $\Delta \geq 0.01$ do
    $t = t + 1$
    Update trustiness of payees using
    $T(v) = \frac{\sum_{u \in \mathrm{In}(v)} R(u) \cdot S(u,v) + ML\_Suspic(v)}{|\,\mathrm{In}(v)|}$
    Update reliablity of payers using
    $R(u) = \frac{\sum_{v \in \mathrm{Out}(u)} T(u) \cdot S(u,v) + ML\_Suspic(u)}{|\,\mathrm{Out}(u)|}$
    $\Delta_T = \sum_{v \in V} |Trust^t(v) - Trust^{t-1}(v)|$
    $\Delta_R = \sum_{u \in U} |Reliab^t(u) - Reliab^{t-1}(u)|$
    $\Delta = \max\{\Delta_T, \Delta_R\}$
end while
Calculate ML Risks of accounts using an equation in [9]
return
```

accounts usually transferred money out only once, while moderate-high-risk and high-risk accounts usually transfer money out many times, since those moderate-risk accounts did not have a strong intention to launder the money actively, the ML risks are relatively low.

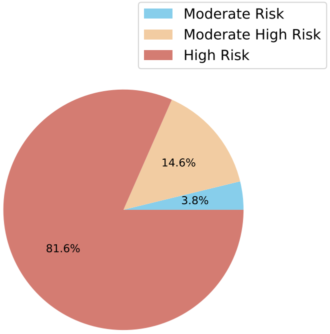

Fig. 1. Risk distribution of accounts in Upbit hack gang. The colors show the different levels of risk.

As shown in Fig. 2, since we crawled transactions layer by layer, we categorized layering accounts into 11 layers and found that as the number of layers increased, the minimum risk of layering accounts decreased. That indicates that layering

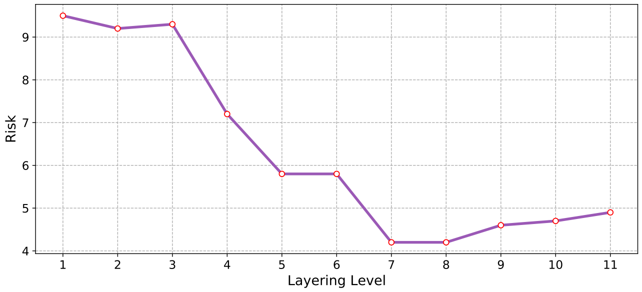

Fig. 2. The minimum risk of different layering level accounts.

transfers of illicit money are indeed a method of risk diversification. It is important to prevent the layering transfer of accounts as early as possible.

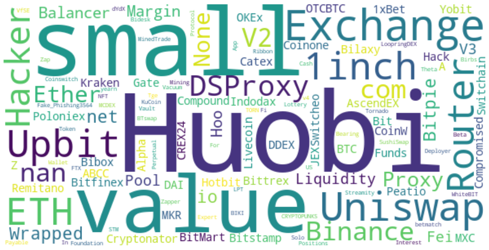

Fig. 3. Service providers of the Upbit hack gang.

The word cloud of service providers in the Upbit hack gang is shown in Fig. 3, we can tell that the stolen money from Upbit hacks flowed into centralized exchanges (such as Huobi and OKEx) and decentralized exchanges (such as Uniswap), and the stolen money also were swapped into other tokens (such as BTC and NFT).

As shown in Fig. 4, we present a local ML process of $G_{refined}$. The entire ML process is a multi-directional chain-like structure, while locally it takes on a diverging tree-like structure, which indicates that the members of the Upbit hack gang were very cunning. Not only did they layer-by-layer transfer the illicit money, but they also dispersed the black money to different accounts, greatly increasing the concealment of criminals.

5 Conclusion

In this paper, we explore the real scale of the Upbit hack gang. We construct a rough suspicious ML transaction network. Then we design an ML suspiciousness indicator for accounts and modify a general risk assessment framework to calculate ML risks of accounts. Based on ML risks, we acquire a more accurate gang of Upbit hacks. In the end, we find that the size of the Upbit hack gang is much

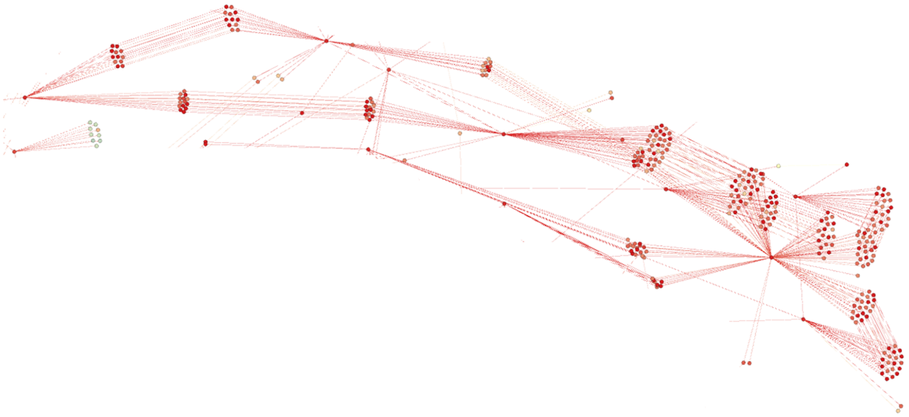

Fig. 4. The portion of ML process of Upbit hack gang. The darker the color of the node, the higher the risk.

bigger than we thought. We also present several interesting findings of ML in the Upbit hack gang.

References

1. Lin, D., Chen, J., Wu, J., Zheng, Z.: Evolution of ethereum transaction relationships: Toward understanding global driving factors from microscopic patterns. IEEE Transactions on Computational Social Systems **9**(2), 559–570 (2022)
2. Wu, J., Liu, J., Zhao, Y., Zheng, Z.: Analysis of cryptocurrency transactions from a network perspective: An overview. Journal of Network and Computer Applications **190**, 103139 (2021)
3. Chen, W., Zhang, T., Chen, Z., Zheng, Z., Lu, Y.: Traveling the token world: A graph analysis of ethereum erc20 token ecosystem. In: Proceedings of The Web Conference. pp. 1411–1421 (2020)
4. Chainalysis Team: The chainalysis 2023 crypto crime report (2023), https://go.chainalysis.com/2023-crypto-crime-report.html
5. Wu, Z., Liu, J., Wu, J., Zheng, Z.: Transaction tracking on blockchain trading systems using personalized pagerank. doi preprint doi:2201.05757 (2022)
6. Kolachala, K., Simsek, E., Ababneh, M., Vishwanathan, R.: Sok: money laundering in cryptocurrencies. In: Proceedings of the 16th International Conference on Availability, Reliability and Security. pp. 1–10 (2021)
7. Sun, X., Feng, W., Liu, S., Xie, Y., Bhatia, S., Hooi, B., Wang, W., Cheng, X.: Monlad: Money laundering agents detection in transaction streams. In: Proceedings of the ACM International Conference on Web Search and Data Mining. pp. 976–986 (2022)
8. Sun, X., Zhang, J., Zhao, Q., Liu, S., Chen, J., Zhuang, R., Shen, H., Cheng, X.: Cubeflow: Money laundering detection with coupled tensors. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 78–90 (2021)
9. Lin, D., Wu, J., Fu, Q., Zheng, Z., Chen, T.: Riskprop: Account risk rating on Ethereum via de-anonymous score and network propagation. arXiv preprint arXiv:2301.00354 (2023)