Finding Cryptocurrency Attack Indicators Using Temporal Logic and Darkweb Data

Mohammed Almukaynizi$^1$, Vivin Paliath$^2$, Malay Shah$^2$, Malav Shah$^2$, Paulo Shakarian$^{1,2}$
$^{1}$Arizona State University
$^{2}$Cyber Reconnaissance, Inc.
Tempe, AZ, USA
{malmukay, vivin, msshah11, mpshah5, shak}@asu.edu

Abstract—With the recent prevalence of darkweb/deepweb (D2web) sites specializing in the trade of exploit kits and malware, malicious actors have easy-access to a wide-range of tools that can empower their offensive capability. In this study, we apply concepts from causal reasoning, itemset mining, and logic programming on historical cryptocurrency-related cyber incidents with intelligence collected from over 400 D2web hacker forums. Our goal was to find indicators of cyber threats targeting cryptocurrency traders and exchange platforms from hacker activity. Our approach found interesting activities that, when observed together in the D2web, subsequent cryptocurrency-related incidents are at least twice as likely to occur than they would if no activity was observed. We also present an algorithmic extension to a previously-introduced algorithm called APT-Extract that allows to model new semantic structures that are specific to our application.

I. INTRODUCTION AND DATASET

Cryptocurrencies are digital currencies that mostly use the blockchain concept to record transactions. Perhaps the most well-known one is Bitcoin. It was estimated that the market capitalization of cryptocurrencies has exceeded 400 billion dollars, after peaking at over 700 billion dollars.

With the high reliance on technology, increasing adoption from businesses and traders, and due to the inherent anonymity associated with transactions and wallet owners, malicious threat actors (including hackers and scammers) aiming for financial gain have been highly motivated to hack and scam to gain control over cryptocurrency wallets and perform transactions. In this research effort, we have encoded and recorded over 50 major incidents of cyberattacks and fraud campaigns targeting multiple cryptocurrencies and traders. A few examples are reported in Table I. We also queried postings from over 400 hacker forums in the D2web using an API that is commercially provided by a threat intelligence firm. The database is collected from D2web sites that were identified to be serving hacking-related content. Using both databases, we mine for sequential patterns to identify indicators of attacks from the D2web database, which can then be monitored to reason about risks that are likely to occur in the future. Specifically, we found that when certain D2web activities are observed together, subsequent cryptocurrency-related incidents are at least twice more likely to occur than they would if no activity was observed and significantly more likely to occur than when individual activity is observed.

### TABLE I

| Name of Attack    | Targeted Currency | Estimated Loss (in USD) |
|------------------|-------------------|-------------------------|
| Mt Gox Hack       | Bitcoin           | 450 Millions            |
| NiceHash Hack     | Bitcoin           | 62 Millions             |
| Parity Wallet Hacks | Ethereum         | 160 Millions            |
| Tether Token Hack | Tether            | 30 Millions             |

To learn such patterns, we sought to derive temporal logic rules of the form “A cryptocurrency $G$ will be targeted by hackers/scammers with a probability $p$ within $\Delta t_{act}$ time units after itemset $F$ is observed on the D2web within $\Delta t_{con}$ time units”. To do so, we combine concepts from logic programming (in particular, the concepts of existential frequency function ($efr$) from APT logic [1], [2]) with frequent itemset mining [3] and temporal causal reasoning [4]. Table II shows some of the interesting $efr$ rules obtained from the algorithm.

II. TECHNICAL APPROACH

A. APT-Logic

### Concepts

We use the same syntax and semantics that were previously introduced in our work [1], [2], [5]. Here, we informally review a subset of those concepts.

A thread is a sequence of worlds (events). Each world corresponds to a discrete time point. Time points are represented by natural numbers in the range $1, \ldots, t_{max}$. A formula $F$ can be any itemset (in this paper, atoms can only be predicates with arities of 2 or 3, e.g., \{attacked(Bitcoin), mentioned(Coinbase, software, 9), mentioned(Gmail, software, 13)\}). The atoms used in this study are partitioned into two disjoint subsets: condition atoms from D2web activities ($A_{con}$), and action atoms from the database of observed incidents ($A_{act}$).

### Satisfaction

We say a thread ($Th$) satisfies a formula $F$ at a time-point $t$ (denoted $Th(t) = F$) if and only if:

$$\forall a \in F (Th(t) = a).$$
Rule probability. The probability is determined based on the fraction of times where a rule r is satisfied from the times where the precondition is satisfied.

Significance. We determine whether a rule is statistically significant based on relative likelihood, i.e., the percentage increase in likelihood of occurrence when the precondition is observed.

B. D2web Activity

Tagging. The D2web API supplies tags with each post. Each tag should belong to one of three categories: financial, software, or general topic tags. The tagging algorithm leverages document similarity techniques on doc2vec representations of posts to assign tags that are most relevant to each post.

Our approach uses the count of tags per day, i.e., when a document similarity techniques on doc2vec representations

Algorithm 1 Forward rolling condition atoms

Input: Thread Th, an empty thread Th’, length of threads t max, condition atoms A con, and length of rolling window ∆t con

Output: Forward-rolled thread Th’

Algorithm 2 Construct Th itemsets

Input: Thread Th, thread Th’, an empty thread Th itemsets, length of thread t max, freqItemsets, and action atoms A act

Output: Th itemsets

utable.

efr Rule-Learning. We use the algorithm APT-EXTRACT to


table.

efr Rule-Learning. We use the algorithm APT-EXTRACT to

mine for efr rules [1]. Essentially, APT-EXTRACT produces all rules whose precondition is any of the itemsets that are satisfied by the thread at \( n \) or more time points, i.e., \( n \) is greater than a minimum (lower bound) support (specified by the argument \( SuppLB \) [1]).

### III. Experiments and Analysis

We show in this section: (1) the rules that are pre-conditioned on itemsets tend to have higher correlations with future incidents than the rules that are pre-conditioned on single atoms, and (2) the performance of our approach change linearly when \( \Delta t_{act} \) is extended.

**Itemset Rules vs. Atomic Rules.** Interestingly, the efr rules whose precondition is itemsets have notably higher probabilities than those whose precondition is single tags. It is evident (from Figure [1]) that the usage of the Apriori algorithm with APT-logic added considerable value to the generated rules.

![Fig. 1. Box plot showing precondition type (itemset rules, single-atom-rules) vs. probability.](image1.png)

Although the single atoms that are used as preconditions of the generated rules may not be in any of the frequent itemsets, the single-atom-rules have on average higher support than the frequent-itemset-rules as depicted in Figure [2]. This suggests that an atom of an itemset does not necessarily occur when another atom in that same itemset is observed i.e., less co-occurrence.

![Fig. 2. Box plot showing precondition type (itemset rules, single-atom rules) vs. support](image2.png)

**Run-time Analysis.** Figure [3] shows that when the length \( \Delta t_{act} \) increases, the time taken for APT-EXTRACT to generate rules increases linearly, because the number of time points for which APT-EXTRACT needs to check the satisfaction of the consequence increases. This shows that our approach is flexible to applications where the user wants to extend the length of time-units for which efr rules are computed. However, we shall note that the Apriori algorithm runs in exponential time, and for all analysis in this study, we restrict the number of items to 2 in each itemset, and we do not introduce more types of atoms (tags).

![Fig. 3. Time taken to generate rules when varying \( \Delta t_{act} \).](image3.png)

### IV. Conclusion

This study presents an approach that combines concepts from causal reasoning, itemset mining, and logic programming. Our approach identifies indicators of cyber threats to cryptocurrency traders and exchange platforms from hacker activity in D2web. Examples of interesting rules are presented in this paper. In the future, we plan to use the approach with other sources such as social media platforms. Additionally, we look to incorporate the learned rules into an alert system that generates and visualizes warnings.

### Acknowledgment

Some of the authors were supported by the Office of Naval Research (ONR) Neptune program. Paulo Shakarian, Vivin Paliath, Malay Shah, and Malav Shah are supported by the Office of the Director of National Intelligence (ODNI) and the Intelligence Advanced Research Projects Activity (IARPA) via the Air Force Research Laboratory (AFRL) contract number FA8750-16-C-0112. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, AFRL, or the U.S. Government.

### References

[1] P. Shakarian, A. Parker, G. I. Simari, and V. S. Subrahmanian, “Annotated probabilistic temporal logic,” ACM Trans. Comput. Logic, vol. 12, no. 2, pp. 14:1–14:44, Jan. 2011.
[2] P. Shakarian, G. I. Simari, and V. S. Subrahmanian, “Annotated probabilistic temporal logic: Approximate fixpoint implementation,” ACM Trans. Comput. Logic, vol. 13, no. 2, pp. 13:1–13:33, Apr. 2012.
[3] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” in ACM sigmod record, vol. 29, no. 2. ACM, 2000, pp. 1–12.
[4] S. Kleinberg and B. Mishra, “The temporal logic of causal structures,” in Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. AUAI Press, 2009, pp. 303–312.
[5] A. Stanton, A. Thart, A. Jain, P. Vyas, A. Chatterjee, and P. Shakarian, “Mining for causal relationships: A data-driven study of the islamic state,” in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015, pp. 2137–2146.