J Vis (2024) 27:469–483
https://doi.org/10.1007/s12650-024-00969-z

REGULAR PAPER

Xuan Chen · Xincan Zhang · Zhaohan Wang · Kerun Yu · Wong Kam-Kwai · Haoyun Guo · Siming Chen

Visual analytics for security threats detection in Ethereum consensus layer

Received: 29 November 2023 / Revised: 29 November 2023 / Accepted: 22 January 2024 / Published online: 18 March 2024

Abstract The Ethereum consensus layer provides the Proof of Stake (PoS) consensus algorithm with the beacon chain for the Ethereum blockchain network. However, the beacon chain is proved vulnerable to consensus-targeted attacks, which are difficult to detect. To address this issue, blockchain developers require an interactive tool to identify and mitigate potential security threats. Currently, most blockchain visualization solutions only display client logs or transaction records, making responding quickly to security threats challenging. This paper introduces the first visual analytics solution for security threat awareness on the Ethereum consensus layer. We cooperate with blockchain experts and investigate a top-down exploration approach, providing an overview of the general security level, as well as detailed consensus achievements in each slot. Our visual system lets users discover specific outcomes of the consensus execution and identify anomalies in the beacon chain historical data. Furthermore, the system includes two case studies of actual attacks to help developers better understand and mitigate potential security threats.

Keywords Ethereum · The consensus layer · The beacon chain · Security

X. Chen (✉) · H. Guo · S. Chen
School of Data Science, Fudan University, Shanghai, China
E-mail: chenxuan@fudan.edu.cn

H. Guo
E-mail: s28@msn.com

S. Chen
E-mail: simingchen@fudan.edu.cn

X. Zhang
School of Economics, Fudan University, Shanghai, China
E-mail: 20307130318@fudan.edu.cn

Z. Wang
School of Computer Science, Fudan University, Shanghai, China
E-mail: 20307130171@fudan.edu.cn

K. Yu
Department of Physics, Fudan University, Shanghai, China
E-mail: 19307110540@fudan.edu.cn

W. Kam-Kwai
Hong Kong University of Science and Technology, HKSAR, China
E-mail: kkwongar@connect.ust.hk

1 Introduction

Since blockchain technology was introduced as a state-of-the-art distributed ledger in 2008 (Nakamoto 2008), it has received a lot of attention from researchers around the globe. Blockchain is a growing list of records, known as blocks. Those blocks are securely linked together via cryptographic hashes with user transactions inside the block body. In a blockchain network, the consensus mechanism selects and generates new blocks. Ethereum is one of the most popular blockchain networks, which contains two components, namely, the execution layer and the consensus layer. The execution layer collects and executes user transactions and packages those transactions into the block body. At the same time, the consensus layer adopts a Proof-of-Stake (PoS) consensus mechanism, which achieves the consensus among nodes in a blockchain network and forms the block header. The beacon chain, as the core component of the Ethereum consensus layer, is designed to produce a block in each 12-second time slot. Users can stake a certain amount of cryptocurrency to become validators and participate in the consensus process. In each slot, a validator is randomly chosen to serve as the proposer, which is responsible for proposing a block for this slot. Other validators have to generate attestations to vote for blocks they verified as correct. Attestations are aggregated and recorded into the beacon chain. Blocks voted by enough attestations, along with the corresponding transactions from the execution layer, are regarded as the canonical chain, which is the unique legitimate Ethereum blockchain.

However, the PoS consensus mechanism of the beacon chain has already been proven vulnerable to consensus targeted attacks (Neuder et al. 2021). First, the blockchain network is considered a partially synchronous network, in which messages are not fully synchronized due to factors such as unintentional network delay or node failure. Attackers could employ network disruption as an opportunity to conduct attacks. Second, it is difficult to identify such attacks instantly. Attackers may launch attestations engaging in appropriate behaviors, which are not recognized as attacks immediately until generating effects. Security threats may be concealed in massive volumes of data, making it difficult to distinguish them from normal

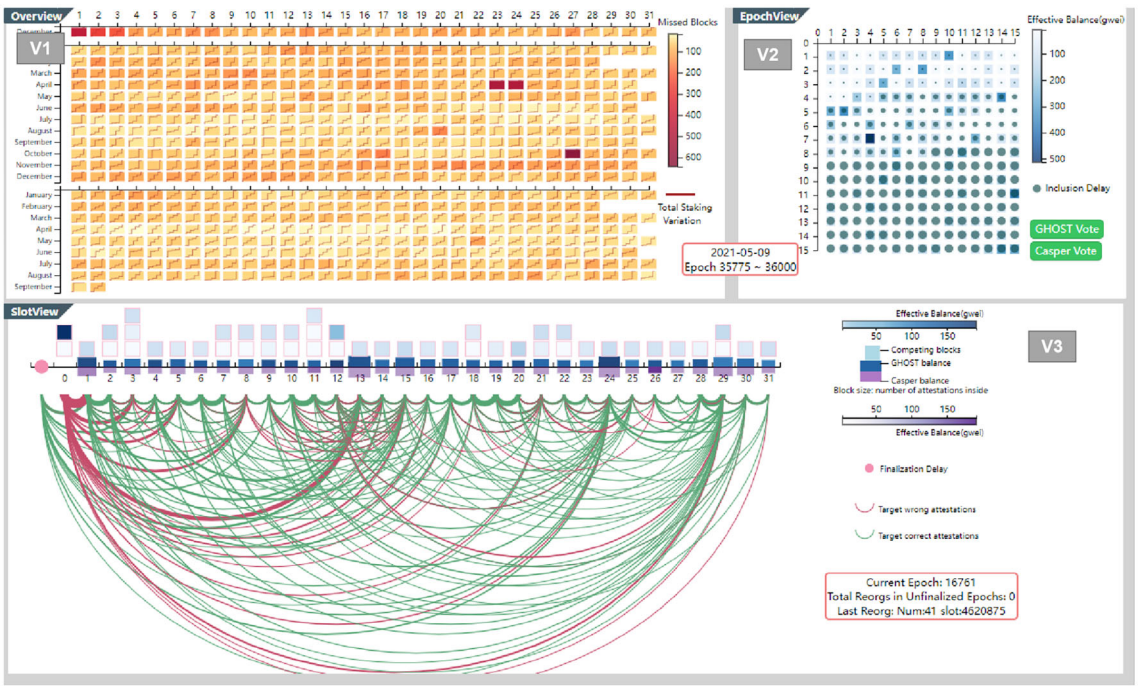

Fig. 1 Screenshot of the visual analytics system. (V1) The overview displays the entire beacon chain data in a calendar format, along with the variation of total deposits, which indicates the general security level of the beacon chain. (V2) The epoch view shows the relevant epoch selected in the overview, reflecting the total effective balance of validators who voted for Casper and GHOST attestations, respectively. The inclusion delay is attached to each epoch, indicating the general security level of consensus execution. (V3) The slot view depicts the execution results of the PoS consensus in a specific epoch. The canonical blocks are shown in the upper panel, together with the number of competing blocks for the slot and the effective balance they have obtained. The results of the checkpoint confirmation are shown below, showing the distribution of attestations and the inclusion delay. By combining these three perspectives, we can detect potential security threats within the beacon chain

operations. Therefore, data analysis may be unable to distinguish between common behaviors and security issues, where expert judgment is required. Meanwhile, visualization is appropriate for connecting human information with security-related data. But as far as we can tell, the majority of the blockchain visualization methods are focused on delivering comprehensive on-chain data. We have not yet discovered a visualization technique for identifying security threats in the Ethereum consensus layer. Blockchain developers usually closely monitor the block recordings to detect irregularities.

This work provides the first visual analytics solution for security threat awareness on the Ethereum consensus layer. Our team members have experience in both blockchain and visualization. We also closely cooperate with domain experts when developing our visual analytics system. Firstly, we investigate the Ethereum communities and previous work and summarize four typical attacks from the most frequently discussed security concerns. Secondly, on the basis of community posts and relevant literature, we outline the four security issues and their potential effects on blockchain data. After which, we extract the requirements, indicating the required data characters when illustrating security threats on the beacon chain. Thirdly, based on those tasks, we design a visual analytics system incorporating three components: overview, epoch view, and slot view. Our visualization system follows a top-down interactive exploration flow. A heat map illustrates the historical epoch status for each day from the initiation of the beacon chain (Fig. 1V1). Operators can select a square on the map to expand the epochs produced on that day in the epoch view (Fig. 1V2). Operators can identify the anomalous epoch with characters highlighted on each square. By choosing a particular epoch, operators can observe the consensus outcomes in the slot view (Fig. 1V3), which emphasizes the distribution of irregular attestations. Our system provides visual analytics methods for inspecting time-series high-dimensional data (Zhao 2023). Based on specific requirements (e.g., security monitoring), we extract relevant data and present it over various time frames. By analyzing different time frames, operators can identify potential security threats and assess risk indicators quickly. Finally, we have conducted two case studies and received recognition from professionals, confirming the usability of our system with historical block data.

The major contributions of this paper are as follows:

1. An application-driven security visual analysis approach of security issues, data character collection, related tasks extraction, and target data visualization in the application of blockchain.
2. The first visual analytics solution for security threat awareness on the Ethereum consensus layer, identifying the visual patterns of security issues, which has been open sourced[1] under the license of MIT.
3. Two security vulnerabilities identified from actual beacon chain data, associated with genuine security incidents with explanations.

The remainder of the paper is organized as follows. The background of the Ethereum consensus layer is discussed in Sect. 2. Section 3 presents related works. We present the security challenges and the visualization design process in Sect. 4. Section 5 presents the detailed descriptions of our system, and Sect. 6 describes two case studies. Section 7 discusses the limitations and the prospects of the proposed system. Finally, this paper is concluded in Sect. 8.

2 Background

Our research is concentrated on the security vulnerabilities of the Ethereum consensus layer. This section introduces the background of the Ethereum consensus layer and the security threats against it.

2.1 Ethereum blockchain

The Ethereum blockchain platform was conceived and implemented in 2013 (Buterin et al. 2014). The Ethereum blockchain previously executed a well-known Proof-of-Work (PoW) consensus mechanism (Wood 2014), known as mining. In order to solve the scalability issue (Zheng et al. 2017), Ethereum introduced the beacon chain with a PoS consensus mechanism in December 2020. Then, in September 2022, the PoW mechanism was abandoned in favor of the PoS mechanism. Meanwhile, the previous blockchain continues to collect and execute user transactions, which is recognized as the Ethereum execution layer. At

[1] https://github.com/AmbitionCX/EthVis.

the same time, the beacon chain provides consensus on new blocks among validators, namely, the Ethereum consensus layer. These two layers collaborate to generate new blocks.

2.2 PoS consensus protocol

In order to participate in and profit from the PoS consensus process, users need to stake a certain amount of cryptocurrency (e.g., 32 Ether) to become a validator. The beacon chain will randomly select a proposer from the validator set for each slot. Proposers are responsible for releasing a block during their slot, while other validators are required to issue attestation for blocks they deem valid. In real-world scenarios, proposers and validators could miss their slots due to factors like network congestion, which brings uncertainty to the PoS consensus process. Therefore, validators need to select a correct block for that slot following the header selection rule (Latest Message Driven Greediest Heaviest Observed SubTree, LMD-GHOST (Buterin et al. 2020)). And selected block will be further finalized as the canonical chain by the confirmation rule (Casper the Friendly Finality Gadget, Casper FFG (Buterin and Griffith 2017)). These are two core components of the Ethereum PoS consensus mechanism. Every validator is expected to produce one attestation per epoch, which comprises two levels of meaning in response to the two algorithms mentioned above.

LMD-GHOST: In the Ethereum network, nodes will receive conflict blocks due to unintentional network failure or malicious conduct, which are recognized as forks. The LMD-GHOST algorithm contributes to the selection of the beacon chain header from a bunch of fork chains depending on their own observations. Specifically, the LMD-GHOST algorithm selects the header blocks from several competing blocks based on recent messages, which are the subsequent blocks constructed based on that one. Blocks with the most number of recent messages attached will be selected as the beacon header, which means the majority of other nodes recognizes the block. Each attestation contains a *header* field, indicating the block hash of its **GHOST vote** result. The block with the most GHOST votes will be considered a canonical block at the end of epochs.

Casper FFG: When an epoch is finished, it will be further finalized by the Casper FFG algorithm. The first block of each epoch is defined as the checkpoint (Also called Epoch Boundary Block in some literature). Each attestation contains a *target* field and a *source* field, corresponding to the current checkpoint and the checkpoints of the preceding epoch, respectively. Such a pair of *target* and *source* fields are named as a **Casper vote** in this work. At the end of each epoch, the checkpoint will be justified if it has collected over $\frac{2}{3}$ of attestations (weighted by the effective balance of each validator), which is considered a *supermajority*. Meanwhile, the immediate previous justified checkpoint of the *source* block becomes finalized. Blocks between justified and finalized checkpoints are considered justified, waiting for further finalization. While blocks between finalized checkpoints are immutable, transactions and attestations in which cannot be modified by any peer. However, in real scenarios, a checkpoint may fail to collect a *supermajority* due to proposer off-line or network congestion, the justification will be delayed, and the epoch will remain to schedule. An example of the PoS consensus mechanism process is illustrated in Fig. 2.

2.3 Security threats towards Ethereum consensus layer

The Ethereum PoS consensus mechanism is elaborated to avoid common attacks on PoS-based systems, for instance, the long-range revisions (Deirmentzoglou et al. 2019) and catastrophic crashes (Buterin and Griffith 2017) due to the combination of two consensus mechanisms. However, it also faces new security threats, especially consensus targeted attacks. Those attacks aim at obstructing the consensus process, where

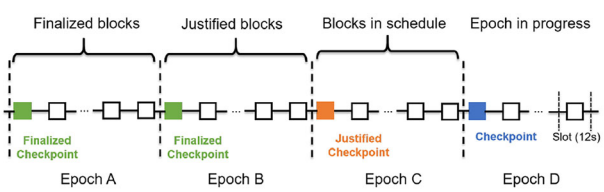

Fig. 2 An example of the Ethereum PoS consensus mechanism. In this diagram, the checkpoint for epoch D will be justified once it receives a *supermajority* at the end of this epoch. Additionally, the previously justified checkpoint (the checkpoint of epoch C) will get finalized. As a consequence, blocks among epoch B that were previously justified will get finalized, whereas unjustified blocks among epoch C will get justified

there are three main realistic behaviors on the beacon chain: reorganizations, finality forks, or finality delays (jmcook.eth 2023). First, the reorganizations (abbreviated as reorg) refer to the replacement or rearrangement of blocks that have previously been settled at the beacon chain head, where attackers can insert their malicious messages or delete any specific messages. Second, depending on the network architecture or propagation latency, validators may have diverse opinions on the beacon chain header, where there will be various fork chains linked to a single block. Most of the time, only one fork chain is ultimately justified and finished. In comparison, a finality fork involves the simultaneous finalization of numerous fork chains. Third, a finality delay prevents the network from gathering a *supermajority*. As a result, the network is unable to satisfy the requirements to finalize or justify any epoch. In the delayed epochs, user transactions cannot be confirmed, and the PoS consensus mechanism is deliberately interrupted. In Sect. 4, we summarize four significant security risks that aim to cause those behaviors.

3 Related work

3.1 Blockchain visualization

Considering the inherent multi-dimensional data produced by blockchain systems, there have been many visualizing works from different perspectives. Intuitive visualizing views of on-chain messages are illustrated on the website. For example, the Bitcoin Big Bang (The Bitcoin Big Bang 2023) visualizes the overtime emergence and interconnectivity of the largest entities on the Bitcoin network. The TxStreet (TxStreet 2023) presents a live memory pool visualization, with incoming transactions represented as passengers seeking to board blocks, which are drawn as buses. Mining activity is one of the most concerning topics in PoW-based blockchain networks. MiningVis (Tovanich et al. 2021) offers intuitive insight into the Bitcoin mining ecosystem, demonstrating mining market statistics, mining pool perl rankings, and pool hopping behaviors. SuPoolVisor (Xia 2020), from a micro perspective, displays the distribution of computational power and reward between different mining pools. BitExTract (Yue et al. 2018) considers Bitcoin transactions. The authors provide an interactive visual analytics solution for exploring evolutionary transaction patterns. Works in Fleder et al. (2015) and Chan and Olmsted (2017) concentrate on the graphical analysis, persisting the transactions and the users' activities into graph database in Bitcoin and Ethereum networks, respectively. A closer look at the users' behavior is presented in BitVis (Sun et al. 2019). The authors present an interactive visualization of Bitcoin account behavior, which allows users to explore by filtering transactions on-demand as well as monitoring Bitcoin-related financial crimes. BitConeView (Di Battista et al. 2015) deploys flow charts to detect financial fraud. Both works pay much attention to security threats.

3.2 Security visualization

Traditional network security practitioners frequently employ the CIA (i.e., Confidentiality, Integrity, and Availability) triangle paradigm to identify vulnerabilities and develop solution strategies (Simmonds et al. 2004; Zhao et al. 2023; Zhou 2023). Most security monitoring systems tend to concentrate on detecting network-related incidents. Authors in Wan et al. (2020) simulate four attacks and capture network data characters. Attacks like DDoS and SQL injection are detected by analyzing TCP packages in Casola et al. (2019). A thorough categorization of network security visualization systems from various user-case perspectives is provided by Shiravi et al. (2012). An interactive visualization solution based on the NVD (National Vulnerability Database) is offered by CVExplorer (Pham and Dang 2018). Situ (Goodall 2019) provides a visual network abnormal detection system, which assists operators in being aware of the current network condition and in spotting potential threats, which mainly keeps track of the accessibility of the Internet.

However, network security challenges in the context of blockchain are distinct from those in more conventional networks and workflows in the business domain (Cheng et al. 2018; Lin 2021; Kam-Kwai et al. 2023). Firstly, due to extensive usage of encryption technologies, data on a blockchain gains significantly higher confidentiality and integrity (Wang et al. 2019). Secondly, the decentralized structure of the blockchain networks ensures higher availability than centralized networks (Samreen and Alalfi 2021). Therefore, the majority of security risks in the blockchain area are related to mechanical flaws or program vulnerabilities. HyperSec (Putz et al. 2021) is a visual analytics security monitoring tool on Hyperledger

Fabric blockchain network. The authors are concerned with four aspects of vulnerabilities, including smart contracts, networks, consensus, and authentication. Different from the permissioned blockchain network, we concentrate on a permissionless blockchain network. We present a visualization approach for the beacon chain security threats. Our visual analytics system focuses on the consensus mechanism, and consensus targeted security risks. To the best of our knowledge, this work is the first study that emphasizes situation awareness for the Ethereum consensus security vulnerabilities.

4 Visual requirements

4.1 Security threats

We summarize four significant security threats which try to attack the PoS consensus procedure. The detailed explanations and their possible impacts are listed as follows:

ST1 **The bouncing attack** (Nakamura 2023a): A bouncing attack utilizes information asymmetries and network congestion between nodes. Attackers who control a subset of validators can execute a bouncing attack at the unfinalized epochs. When two distinct fork chains exist in the current blockchain network (for example, fork A and fork B). Assume the checkpoint of fork A is on the verge of gathering a *supermajority*. Attackers can leverage the attestations on fork B and release attestations via assaulting validators. As a result, fork B amass a *supermajority* before fork A, which fails to be finalized. Transactions in the corresponding execution layer are discarded. The attackers successfully reverse the **Casper vote** results, leading to a short-range reorganization of the beacon chain. In this process, attackers can force a fabricated block with illegal transactions to be finalized into the canonical chain or prevent transactions in fork A from being finalized (Nakamura 2023b).

ST2 **The balance attack** (Neu 2023a, b): Similar to the bouncing attack, attackers purposefully utilize the messages propagation delay (e.g., unintended or hostile network failure) to divide honest validators into distinct groups. A malicious validator (which will be slashed and ejected from the network later) proposes two conflict blocks as the checkpoints (Neuder et al. 2021). Those conflict blocks will intentionally propagate to different groups of honest validators, which will build subsequent fork chains based on different checkpoints. The remaining malicious validators selectively release the attestations to equivocate on two fork chains to guarantee that none of the checkpoints can amass a *supermajority* for justification. Therefore, the functioning of the beacon chain is terminated with justification delay since the network is unable to reach a consensus on the checkpoint.

ST3 **The avalanche attack** (Joachim Neu 2023): The avalanche attack takes advantage of a flaw in the PoS protocol's fork-choice LMD-GHOST algorithm. Each validator maintains a table containing the latest message (e.g., proposed blocks) from other validators. The fork choice algorithm counts the number of the latest message attached to the block in each slot to determine which block is legitimate. Attackers manipulate the proposer to propose a large number of the latest message attached to a fabricated block in order to mislead the LMD-GHOST algorithms. The client executing the fork choice algorithms will consider the fabricated block as the most active one and include it in the canonical chain permanently. As a result, attackers can repeat this process to manipulate a long chain of blocks and consequently restructure the beacon chain.

ST4 **The saving attack** (Otsuki et al. 2021): The saving attack is another security threat conducted with network congestion, aiming at reorganization and finality delay. Different from the bouncing attack, the saving attack seeks to repeatedly reverse the justification trends between fork chains. Assume honest validators will vote for the fork chain with more cumulative attestations. Attackers will save their attestations and leverage each fork chain as long as the attestation from honest validators is insufficient for a justification. At a proper time, attackers will release their votes to the fork with fewer attestations gathered until it surpasses another fork chain. Attackers repeat this process, and the beacon chain will keep reorganizing, making it impossible to preserve anything along the chain.

4.2 Data characters

We have had close communication with experts and independent researchers who have experience in blockchain security and have been involved in the design of the beacon chain. Our aim was to understand the security threats and their potential impact on the beacon chain data. Attacks, as we explained in Sect. 2,

could cause noticeable irregularities in block contents or trigger the protection mechanism of the beacon chain. The following are possible data characters (DC) of the beacon chain data and their relevant attacks:

DC1 **Consensus failure**: Consensus failure refers to the Ethereum network failing to reach a consensus in a slot. Consensus failures can occur for various reasons, including the absence of a proposer or propagation delay. It is common if a few missed blocks appear during consensus processes. However, a significant number of missed blocks results in an abnormal situation on the beacon chain, which may be brought on by malicious attacks (ST1, ST4).

DC2 **Finalization or justification delay**: Blocks in the Ethereum consensus layer are justified or finalized epoch by epoch. Nevertheless, it will be delayed when the checkpoint fails to collect a *supermajority* at the end. As a result, the relevant epoch cannot be included in the canonical chain as scheduled. Even though justification or finalization may take place during regular consensus processes, they are still closely related to consensus targeted attacks because they may be a consequence of successfully conducted attacks (ST2, ST4) or a necessity for some attacks (ST1).

DC3 **Insufficient attestations**: Moreover, the checkpoint lacks *supermajority* is also an immediate reason for finalization or justification delay. Specifically, the checkpoints are unable to collect attestations, weighted by effective balance, over $\frac{2}{3}$ of all active validators. A successful attack with the beacon chain reorganizations (ST2, ST4) may result in a substantial fraction of negative attestations. As a result, the percentage of positive attestations is critical for security monitoring.

DC4 **Concentratively distributed attestations**: Attackers often need to maintain control over a subset of malicious validators in order to initiate assaults by releasing their attestations at strategic times. Attackers will need to continuously monitor the on-chain environment, seeking opportunities to reverse (ST1) or mislead other honest validators (ST3, ST4). We can observe certain attestations that are concentratively distributed throughout several slots, voting to a single slot simultaneously.

DC5 **Long inclusion delay**: Instead of unleashing malicious attestations in a short time, attackers can reserve their controlled validators and disseminate them at a suitable time. A long inclusion delay may indicate that nodes encounter poor peer-to-peer communication latency. Additionally, in some attacks (ST1, ST2), attestations will experience a long delay from emission to inclusion since attackers need to hold those attestations and release them at the opportune moment. Therefore, a long inclusion delay may indicate attacks or attempted assaults.

DC6 **Competing blocks for a single slot**: Since the beacon chain selects the fork chain with the LMD-GHOST algorithm, attackers could utilize a large number of fictitious blocks to trick honest validators, such as the avalanche attack (ST3). Those fake blocks will be recorded on the blockchain if they are voted by any attestations. Therefore, there may be a number of competing blocks under the assault, which often share a sizable portion of attestations.

The data characters mentioned above are produced during the consensus process and stored on the blockchain, which we can obtain from a full Ethereum node. In this work, we do not consider reorganization and finality forks as data characters because they are not a part of the beacon chain data. Table 1 summarizes the aim of the assaults and their potential data characters on the beacon chain.

4.3 Requirements analysis

Considering expert feedback, we compile a list of requirements to identify security threats and risks with the aforementioned data characters. We summarize the concrete visualization requirements as follows:

R1 **Understand the general status of epochs and slots.** In order to analyse the security threats among beacon chain data, it is necessary to have a general understanding of the present and historical state. The status of the beacon chain contains many different aspects of data. We focus on the consensus process-related data, including the number of consensus failures (DC1, DC2), the historical

Table 1 Proposed attacks and their impacts on the beacon chain

Security Threats	Data Characters	Reference
ST1: Bouncing Attack	DC1, DC2, DC4, DC5	Nakamura (2023a, 2023b)
ST2: Balance Attack	DC2, DC3, DC5	Neu (2023a, 2023b)
ST3: Avalanche Attack	DC4, DC6	Joachim Neu (2023)
ST4: Saving Attack	DC1, DC2, DC3, DC4	Otsuki et al. (2021)

competing blocks, and the incorrectly voted attestation (DC3), which are significant indicators of the overall security level.

R2 **Browse the final attestation allocation for potential organized attacks.** Attestations are the foundational element of the Ethereum consensus layer. The distribution and the allocation of attestations, including their emission and inclusion slots, are crucial to detect security threats. The specific allocation of attestations will show particular characteristics, for example, concentrating on a specific time period (DC4) or long inclusion delay (DC5).

R3 **Summarize manipulative behaviors to hint at Casper voting.** An attestation is weighted by effective balance, which is correlated to the staking balance of the validator. Therefore, to prevent achievements of justification or finalization, attackers may manipulate the Casper votings, such as excluding certain attestations or cutting off attestation transmission with certain behaviors (DC4, DC5). These actions will lead to justification or finalization delay due to insufficient effective balance (DC3).

R4 **Identify GHOST voting results of each slot.** A GHOST voting is intended to choose canonical blocks for each slot. Attackers can control malicious validators to manipulate GHOST votes, such as swaying the trends of justification or mass-producing malicious messages with specific behaviors (DC4, DC5). Blocks under attack may be forked, leaving the slots unfilled and abandoning attestations from honest validators (DC6).

5 Visual analytics system

Based on the design requirements, we introduce our visual analytics system in this section. As Fig. 3 demonstrates, our system is based on a full Ethereum node. Teku is one of the most widespread Ethereum consensus layer clients. Geth is the official client for the Ethereum execution layer. To extract necessary data, we install chaind, an open-source tool that reads historical block data, reorganizes the attestations, and stores them in PostgreSQL. Finally, with those data, our system is visualized with D3.js and Vue.js.

Our visualization system consists of three views: overview, epoch view, and slot view. Table 2 provides a summary of relationships between data characters and the corresponding requirements, as well as visualization views to represent different requirements. To better explain our data source, we illustrate the beacon block structure and how we extract necessary data from blocks, as shown in Fig. 4. A beacon block consists of three parts: block header, block body, and the user transactions space reserved for the Ethereum execution layer. The block header maintains necessary information about the block itself, whereas the block body stores aggregated attestations submitted by validators, as well as committee description, validator status, etc. Our visualization system is expanded following the time scale, allowing users to interactively explore security vulnerability from macro to micro perspectives.

5.1 V1: Overview

The entrance of this visualization system is the overview (Fig. 1 V1), which serves as an overall description of all historical beacon chain blocks in a calendar format.

Visual Encoding: The beacon chain is allocated 12 s, and an epoch with 32 slots inside is produced in 6.4 min. Therefore, the Ethereum consensus layer generates 225 epochs per day. In the overview, we use a

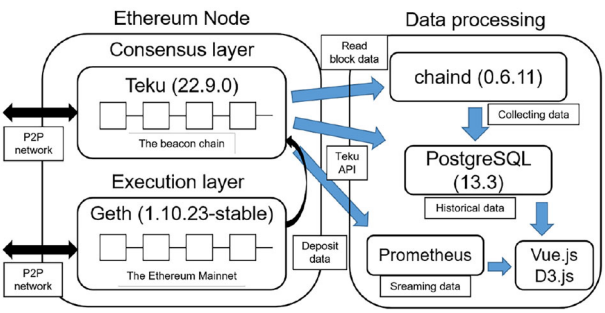

Fig. 3 System structure and data processing of our visualization system

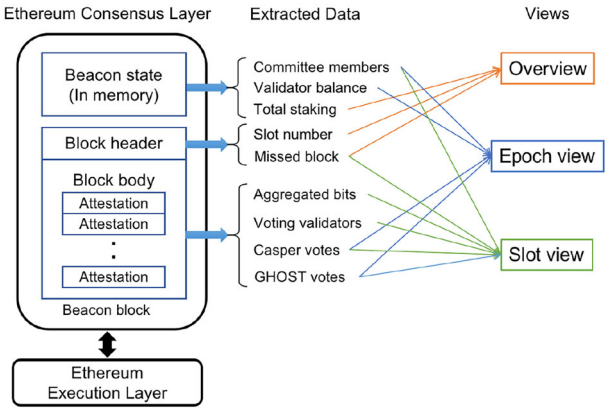

Fig. 4 Data extraction and allocation from beacon blocks to corresponding views

Table 2 Visual requirements and the corresponding views to present data characters

Data Characters	Requirements	Views
DC1	R1	V1, V2
DC2	R1	V2
DC3	R1, R3	V2, V3
DC4	R2, R3, R4	V3
DC5	R2, R3, R4	V3
DC6	R4	V3

rectangle to represent 225 epochs produced each day and arrange these rectangles in a calendar. The calendar starts from December 1st, 2020. A block is missed if the block root is empty. We collect the number of active validators in each epoch and calculate the total deposits. The filling color in each rectangle represents the number of missed blocks in associated epochs. The quantity of missed blocks can serve as a signal to alter irregularities. Another significant factor for the overall security level of PoS consensus is the total amount of deposits (Buterin and Griffith 2017). The more validators engage in the PoS consensus process, the more difficult it is to deceive honest validators, and the system achieves a higher level of security.

Interaction: In the overview, we provide the general level of safety for each day. As a consequence, rectangles in the overview can act as a filter, enabling analysts to select a particular day of interest. By clicking a rectangle in the overview, they can select 225 epochs generated on a corresponding day. The selected date and epoch number are displayed on the right, which initiates the subsequent view to examine the details of the condition of selected epochs.

5.2 V2: Epoch view

Visual Encoding: The 225 epochs on a specific day is expanded in the epoch view. We extract the effective balance-related data from the block body to show the epoch's security. In total balances, we determine the proportion of voted validators. We display the detailed consensus voting results in a matrix from left to right and top to bottom as Epoch view (Fig. 1 V2). After an epoch is finalized or justified, the **Casper vote** and **GHOST vote** of every voted attestation will be denoted as correct or incorrect, indicating whether the checkpoint or head they voted on is included in the canonical chain. The color filled in finalized and justified epochs represents the total effective balance (e.g., the balance of validators with incorrect and unvoted attestations). Meanwhile, the scheduling and delayed epochs are filled with grey since no result has been obtained. Additionally, we calculate the total epoch delay on each epoch circle, indicating the accumulated epoch delay on that day.

The **Casper vote** related effective balance is denoted with blue, while the **GHOST vote**-related one is presented in purple. We highlighted epochs with a high percentage (e.g., epochs with over 20% of **Casper vote**, and over 25% of **GHOST vote**.) An epoch that has been subjected to a bouncing attack may have a percentage of less than one-third non-positive **Casper vote**, where assailants tend to undertake a low-cost

attack. Moreover, practically a balance attack is always preceded by half of both **Casper vote** and **GHOST vote**.

Interaction: Users can choose to observe the effective balance distribution of **Casper vote** or **GHOST vote** results with the green button. The numerical of both voting results, as well as the epoch delay, is shown in a floating window. Besides, users can click one rectangle to expand the slot view of a specific epoch.

5.3 V3: Slot view

Visual Encoding: By clicking a square in the epoch view, users can unfold the voting results of the relevant epoch to the slot view (Fig. 1 V3). We construct the slot view based on the slot timeline, and canonical blocks are filled into each slot. One attestation contains two voting results: the Casper vote and the GHOST vote. Attestations in the same committee with the same opinion are grouped into a bit array, which is designed to reduce the block size. We process all aggregated attestations from the block body and retrieve the voting results of validators. *Target* and *header*, which match the correct canonical block, are recognized as a correct vote, weighted by the number of aggregated validators. Otherwise, it is an incorrect vote, including mismatched and unvoted validators.

In the slot view, the distribution of attestations in each slot is presented by the size of the square, which is encoded with the number of attestations it contains. The upper area displays the **GHOST vote** result, and the lower area depicts the **Casper vote** allocation. Specifically, in the upper GHOST area, we summarize the competing blocks, which have received **GHOST vote** but failed to be recognized as canonical blocks. The intensity of the filling color of the upper canonical block represents the amount of effective balance approving that block. The blocks above the canonical one are the competing blocks, also filled with the effective balance they received, corresponding to the effective balance of incorrect GHOST votes in the epoch view. The effective balance received by the competing block reflects the disagreement level between validators in different fork chains. Similarly, in the lower Casper section, the canonical block is filled with the amount of correctly voted effective balance. We employ an arc diagram to show the Casper voting result in an aggregated attestation. The *target* field in each attestation indicates the selected checkpoint, chosen from the perspective of each node on the current epoch structure. In slot view, correctly voted attestation arcs are green, whereas incorrectly voted attestation arcs are red. The thickness of the arcs represents the number of attestations with the same votes. The arc span, which indicates the inclusion delay, is also an essential expression. Attestations are submitted at the end of arcs and then incorporated into blocks at the other end, resulting in a delay in attestation inclusion.

6 Case study

In this section, we present two case studies with real accidents on the beacon chain. According to characters observed from different views, operators can determine the possible type of threats in Table 1. The workflow for assessing the type of possible attack is depicted in Fig. 5.

6.1 Case 1

Irregularity recognition: The first case on real beacon chain data is on April 23 and April 24, 2021. The beacon chain data had two noticeable irregularities. There were a lot of missed blocks and delayed epochs throughout those two days. We can examine the detailed epochs created on that particular day and show our visual analysis process in Fig. 6.

We can expand the relevant epochs in the epoch view by clicking that day in overview (Fig. 6A). We provide the Casper and GHOST views of the corresponding epoch created on April 23 (Fig. 6B, C) with floating tables. We can observe that there are some rectangles highlighted with a boundary, indicating that the fraction of the voted effective (`DC3`) balance of those epochs is over the warning line (20% in our work). We can identify numerous missed blocks (`DC1`) from epoch 32569 to epoch 32575 by searching around the epoch delay. Moreover, we do not observe any obvious abnormality in the GHOST voting view Fig. 6C. In addition, we display the inception of the irregularity in epoch 32556 (Fig. 6D) and its right next epoch 32557 (Fig. 6E). Above the missed blocks, we can observe that the competing blocks received substantial effective balances, indicating that the LMD-GHOST algorithm works well. However, those blocks still fail

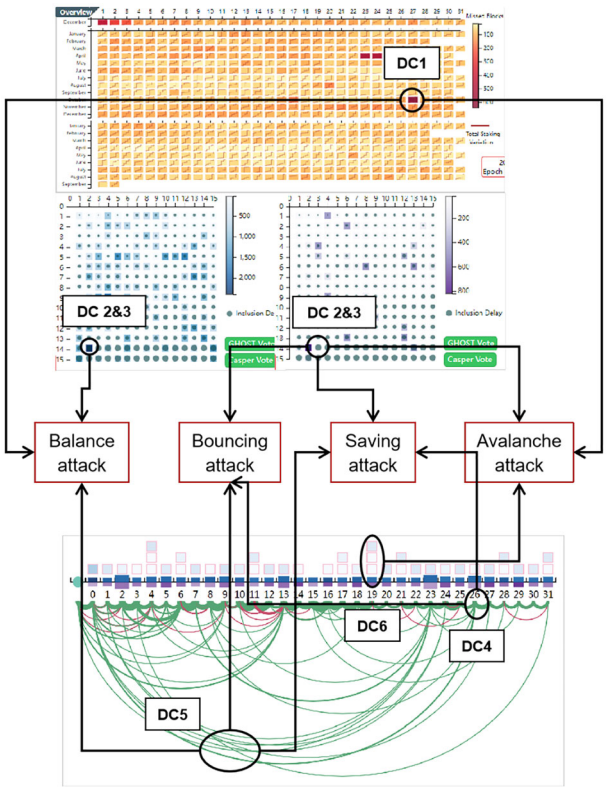

Fig. 5 The procedure for assessing attack types according to data characters from different views. This workflow exemplifies how our system collaborates with several perspectives to evaluate and identify potential threats. For example, if operators simultaneously observe the abnormality in the epoch view, where there are many incorrect GHOST and Casper votes, and the abnormality in the slot view, where some attestations are concentratedly distributed with long inclusion delay. This epoch has a high probability of being attacked by a bouncing attack

to become canonical ones, as we mentioned in Sect. 2, indicating that the proposer failed to disseminate blocks to other peers.

Attack identification: In this case, we observe the characters of competing blocks take a considerable amount of effective balance and missed blocks via our visualization system. There is no matching type of attack according to Fig. 5. Meanwhile, considering we did not find conflicting attestations or blocks on the visualization system, we can conclude that the irregularity between epochs 32302 and 32319 is approximately not a malicious attack.

Case Verification: We verify the irregularity that happened on April 23, 2021, on the beacon chain with experts. The beacon chain was affected by a bug[2] from a popular Ethereum consensus layer client Prysm. The bug first emerged at epoch 32302 and then vanished at epoch 32320. Furthermore, it reappeared on the following April 24. The block proposal functionality was disabled as a result of this problem. Because Prysm is a widespread client, numerous proposers were unavailable during that period. However, proposals with other clients were regular on the other hand. As a result, only a small number of proposers can continue producing canonical blocks. Additionally, since the bug does not affect Casper voting, validators in every slot can submit attestation as usual, which were included in canonical blocks.

6.2 Case 2

Irregularity recognition: An actual accident on the beacon chain happened on May 25, 2022, when a seven-block reorganization occurred at epoch 121471. As we discovered from the overview, epoch 121471 has a considerably darker color of Casper votes (Fig. 7B) than other epochs on May 25, 2022. From the epoch view, we can see epoch 121471 gains a considerable amount of ineffective GHOST voting, indicating

[2] https://medium.com/prysmatic-labs/eth2-mainnet-incident-retrospective-f0338814340c.

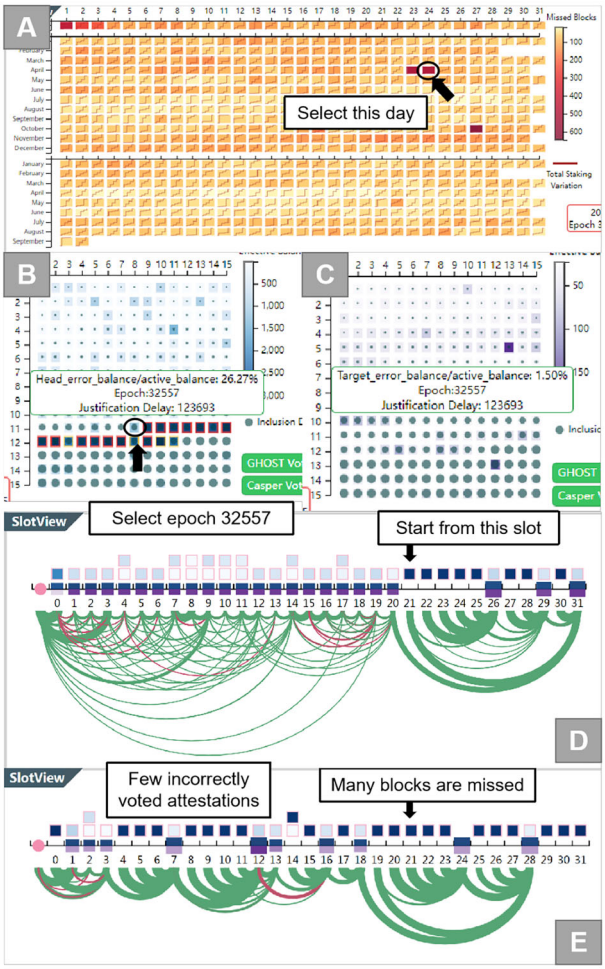

Fig. 6 Case study on the real beacon chain data: visualization of the accident that happened on April 24, 2021 (**A**) Select the specific day on the overview, where the color of this rectangle shows the abnormality on this day (**B**) The Casper voting results on April 24, 2021. The irregularity begins around epoch 32557 and is highlighted by the red border. (**C**) The GHOST voting results on April 24, 2021. (**D**) The slot view of epoch 32556. (**E**) The slot view of epoch 32557

that a lot of attestations are invalid in this epoch. From the slot view, we can see that the seven consecutive slots are empty. These slots have two competing blocks above, which have received a considerable amount of attestations. Second, blocks following missed slots have sufficient attestations. Although they have many competing blocks, those competing blocks do not receive enough attestations. Third, attestations emitted from those missed slots are all invalid with a long inclusion delay.

Attack identification: From those characters, we can observe that the beacon chain was unable to establish consensus in those slots (**DC1**), causing a long inclusion delay (**DC5**) that was resolved 20 slots later. According to Fig. 5, a possible type of attack is the bouncing attack. However, those failures did not trigger a finalization or justification delay, and attestations were also sufficient, even though some validators cast their votes for different targets. Our system indicates that this is not a malicious assault. Additionally, the specialists informed us that this situation is remarkable due to the block reorganization.

Case Verification: This seven-block reorganization is verified with experts and the beacon chain browsers.[3] This is caused by an upgrade of the beacon chain client.[4] The validators with different versions of the client have different views on the weight of the attestation. As a result, blocks proposed by new version clients are not recognized by the old clients. Thus, the attestations included in those blocks are propagated again and judged as incorrect later. In this accident, validators are separated into two groups by different

[3] https://beaconcha.in/epoch/121471.

[4] https://github.com/ethereum/consensus-specs/pull/2878.

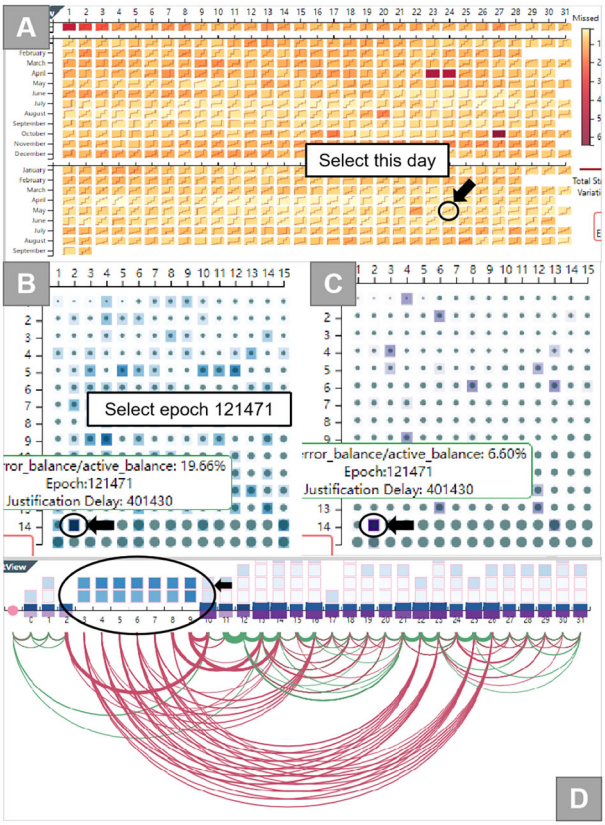

Fig. 7 Case study on the real beacon chain data: visualization of the accident that happened on May 25, 2022 (**A**) The specific day on the overview (**B**) The Casper voting results on May 25, 2022. We can see the abnormal epoch has a darker color than other epochs. (**C**) The GHOST voting results on May 25, 2022. Both GHOST and Casper voting results are abnormal. (**D**) Slot view of epoch 121471. We can see a continuous seven blocks are missed in this epoch

client versions, and these two groups of validators fork the beacon chain. Noticeably, this reorganization did not result in finality or justification delay.

7 Discussion

In this paper, we concentrate on the visual analysis of the Ethereum consensus layer and its PoS consensus protocol. Here are the limitations and future work of our work.

Data limitation: As we discussed in Sect. 4, our data is extracted from a node in the blockchain P2P network, which is commonly modeled with adversarial network delay. In a P2P network, nodes are not expected to be fully synchronized with all messages emitted by all other peers. Therefore, even in ideal network circumstances, message synchronization is also constrained by the information propagation latency. In the real network environment, our system cannot catch all possible disseminating messages among the network, which is the major limitation of our system. To minimize this constraint, we construct our beacon chain client on a cloud server with a reliable internet environment and try to establish connections with as many peers as possible.

Limited aspect of threats: In this work, we only focus on consensus-related security threats and analyse only a small number of attacks. Our system can detect vulnerabilities in the consensus process. However, the Ethereum network faces various security risks from different aspects, such as code vulnerabilities in smart contracts or the privacy problem in front-end applications. While a front-end application vulnerability has no bearing on the blockchain, a smart contract vulnerability is legitimate from a blockchain perspective. These security risks are widespread throughout blockchain networks and are challenging to analyse via visualization.

8 Conclusion

The beacon chain, which serves as the fundamental component of the Ethereum consensus layer, has been proven to be vulnerable to various consensus targeted security threats. As far as we know, the Ethereum consensus layer lacks an intuitive visualization tool for detecting security vulnerabilities. This work provides the first visual analytic solution on security threats for the beacon chain. We cooperate with domain experts in investigating security threats towards the beacon chain. Then extract data characters and visualization requirements, which provides an application-driven security analysis approach. Our visual analytics system consists of three views, which allow operators to interactively detect signals of insecurity in historical blocks and spot potential security threats. In addition, we perform two case studies of actual incidents on the beacon chain. Our system is capable of locating these vulnerabilities and investigating their specific causes in detail.

References

Buterin V, et al. (2014) A next-generation smart contract and decentralized application platform. white paper, 3(37):2–1

Buterin V, et al. (2020) Combining ghost and Casper. arXiv:2003.03052

Buterin V, Griffith V (2017) Casper the friendly finality gadget. arXiv:1710.09437

Casola V, De Benedictis A, Riccio A, Rivera D, Mallouli W, de Oca EM (2019) A security monitoring system for internet of things. Internet of Things 7:100080

Chan W, Olmsted A (2017) Ethereum transaction graph analysis. In: International conference for internet technology and secured transactions (ICITST), pp. 498–500. IEEE

Cheng S, Zhong W, Isaacs KE, Mueller K (2018) Visualizing the topology and data traffic of multi-dimensional torus interconnect networks. IEEE Access 6:57191–57204

Deirmentzoglou E, Papakyriakopoulos G, Patsakis C (2019) A survey on long-range attacks for proof of stake protocols. IEEE Access 7:28712–28725

Di Battista G, Di Donato V, Patrignani M, Pizzonia M, Roselli V, Tamassia R (2015) Bitconeview: visualization of flows in the bitcoin transaction graph. In: Proc. VizSec, pp. 1–8. IEEE

Fleder M, Kester MS, Pillai S (2015) Bitcoin transaction graph analysis. Computer Systems Security

Goodall JR et al (2019) Situ: Identifying and explaining suspicious behavior in networks. IEEE Trans Visual Comput Gr 25(1):204–214

jmcook.eth (2023) Ethereum PoS Attack and Defense. https://mirror.xyz/jmcook.eth/YqHargbVWVNRQqQpVpzrqEQ8IqwNUJDIpwRP7SS5FXs. (Online; Accessed)

Joachim Neu DT (2023) Ertem Nusret Tas. Avalanche Attack on Proof-of-Stake GHOST. https://ethresear.ch/t/avalanche-attack-on-proof-of-stake-ghost/11854. (Online; Accessed)

Kam-Kwai W, Wang X, Wang Y, He J, Zhang R, Qu H (2023) Anchorage: Visual analysis of satisfaction in customer service videos via anchor events. IEEE Transactions on Visualization and Computer Graphics

Lin Y et al (2021) Taxthemis: Interactive mining and exploration of suspicious tax evasion groups. IEEE Trans Visual Comput Gr 27(2):849–859

Nakamoto S (2008) Bitcoin: A peer-to-peer electronic cash system. Decentralized Business Review, p. 21260

Nakamura R (2023a) Analysis of bouncing attack on FFG. https://ethresear.ch/t/analysis-of-bouncing-attack-on-ffg/6113. (Online; Accessed)

Nakamura R (2023b) Prevention of bouncing attack on FFG. https://ethresear.ch/t/prevention-of-bouncing-attack-on-ffg/6114. (Online; Accessed)

Neu J (2023a) A balancing attack on Gasper. https://ethresear.ch/t/a-balancing-attack-on-gasper-the-current-candidate-for-eth2s-beacon-chain/8079. (Online; Accessed)

Neu J (2023b) Attacking Gasper without adversarial network delay. https://ethresear.ch/t/attacking-gasper-without-adversarial-network-delay/10187. (Online; Accessed)

Neuder M, Moroz DJ, Rao R, Parkes DC (2021) Low-cost attacks on ethereum 2.0 by sub-1/3 stakeholders. arXiv:2102.02247

Otsuki K, Nakamura R, Shudo K (2021) Impact of saving attacks on blockchain consensus. IEEE Access 9:133011–133022

Pham V, Dang T (2018) Cvexplorer: Multidimensional visualization for common vulnerabilities and exposures. In: IEEE International Conference on Big Data (Big Data), pp. 1296–1301

Putz B, Böhm F, Pernul G (2021) Hypersec: Visual analytics for blockchain security monitoring. In: IFIP International Conference on ICT Systems Security and Privacy Protection, pp. 165–180. Springer

Samreen NF, Alalfi MH (2021) A survey of security vulnerabilities in ethereum smart contracts. arXiv preprint arXiv:2105.06974

Shiravi H, Shiravi A, Ghorbani AA (2012) A survey of visualization systems for network security. IEEE Trans Visual Comput Gr 18(8):1313–1329

Simmonds A, Sandilands P, Ekert LV (2004) An ontology for network security attacks. In: Asian applied computing conference, pp. 317–323. Springer

Sun Y, Xiong H, Yiu SM, Lam KY (2019) Bitvis: An interactive visualization system for bitcoin accounts analysis. In: 2019 Crypto Valley conference on blockchain technology (CVCBT), pp. 21–25. IEEE

The Bitcoin Big Bang (2023) https://info.elliptic.co/hubfs/big-bang/bigbang-v1.html. (Online; Accessed)

Tovanich N, Soulié N, Heulot N, Isenberg P (2021) Miningvis: Visual analytics of the bitcoin mining economy. IEEE Trans Visual Comput Gr 28(1):868–878
TxStreet (2023) https://txstreet.com/v/eth. (Online; Accessed)
Wang R, Liu H, Wang H, Yang Q, Wu D (2019) Distributed security architecture based on blockchain for connected health: Architecture, challenges, and approaches. IEEE Wireless Commun 26(6):30–36
Wan Y, Xu K, Xue G, Wang F (2020) Iotargos: A multi-layer security monitoring system for internet-of-things in smart homes. In: INFOCOM Conference on Computer Communications, pp. 874–883. IEEE
Wood G et al (2014) Ethereum: a secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper 151(2014):1–32
Xia J-Z et al (2020) Supoolvisor: a visual analytics system for mining pool surveillance. Front Inf Technol Electr Eng 21(4):507–523
Yue X, Shu X, Zhu X, Du X, Yu Z, Papadopoulos D, Liu S (2018) Bitextract: Interactive visualization for extracting bitcoin exchange intelligence. Trans Visual Comput Gr 25(1):162–171
Zhao Y et al (2023) ASTF: visual abstractions of time-varying patterns in radio signals. IEEE Trans Visual Comput Gr 29(1):214–224
Zhao Y, Lv S, Long W, Fan Y, Yuan J, Jiang H, Zhou F (2023) Malicious webshell family dataset for webshell multi-classification research. Visual Informatics
Zheng Z, Xie S, Dai H, Chen X, Wang H (2017) An overview of blockchain technology: Architecture, consensus, and future trends. In: International congress on big data (BigData congress), pp. 557–564. IEEE
Zhou J et al (2023) Dpviscreator: Incorporating pattern constraints to privacy-preserving visualizations via differential privacy. IEEE Trans Visual Comput Gr 29(1):809–819