Looking for Anomalies in Cross-Chain Bridges

André Augusto[1] Rafael Belchior[1,2] Jonas Pfannschmidt[2] André Vasconcelos[1] Miguel Correia[1]
[1]*INESC-ID, Instituto Superior Técnico, Universidade de Lisboa* – Lisboa, Portugal
[2]*Blockdaemon* – Dublin, Ireland
{andre.augusto, rafael.belchior, andre.vasconcelos, miguel.p.correia}@tecnico.ulisboa.pt jonas@blockdaemon.com

I. Introduction

In recent years, there has been a growing adoption of blockchain interoperability solutions and cross-chain protocols [1], [4]. The most popular are cross-chain bridges or, simply, bridges. Bridges connect decentralized applications across various blockchains, supporting the transfer and exchange of assets between blockchains. Cross-chain bridges now have tens of billions of total value locked (TVL). However, this growth has also led to the theft of billions in cross-chain protocols [1]. Not even extensively audited bridges are immune to vulnerabilities [6].

Some authors have studied cross-chain security, systematizing vulnerabilities and attacks across the relevant cross-chain layers [1], [7], [8], [13]–[15]. However, *quantitative studies* with real-world data are still lacking. Furthermore, due to the associated challenges, there is a lack of *security mechanisms* to protect bridges from such attacks. Variations in contract implementations, security models [1], bridging models [2], and token types across different chains make it difficult to monitor and safeguard these systems consistently. Furthermore, the use of intermediary protocols (e.g., bridge aggregators [10], [12]) and the extraction of data from various sources (e.g., transaction data or events emitted by contracts) increase the technical challenges of performing these studies.

We introduce *XChainWatcher*, the first open anomaly detection mechanism for cross-chain bridges, capable of detecting known attacks and other anomalies that harm users and protocol operators. XChainWatcher provides the entire pipeline for extracting cross-chain data from blockchains, decoding data, building logic relations, and evaluating the data against a set of anomaly detection rules.

XChainWatcher was designed and evaluated based on the first empirical analysis of the security of cross-chain bridges. We perform an anomaly detection on data extracted from bridge contracts deployed on Ethereum, Gnosis and Moonbeam. Overall, we analyzed transfers of tokens that collectively moved more than 4.2 billion USD. Through the analysis of the data extracted from blockchains, we identify five new anomalies in cross-chain protocols. Not only can we identify past attacks, but we can also identify unintended behavior from protocols that harm the users.

We model cross-chain operations by establishing a comprehensive set of logical relations that capture events emitted by smart contracts and static configurations common to bridge protocols. We derived them by thoroughly reviewing the open-source code of cross-chain bridge protocols that connect Ethereum to sidechains, and their documentation. We also interacted directly with the developers of some bridges and observed the different state changes. These bridges were Polygon, Ronin, Omnibridge, xDAI Bridge, and Nomad Bridge. These bridges connect Ethereum to multiple sidechains, such as Ronin, Gnosis, Polygon, and Moonbeam.

II. XChainWatcher

XChainWatcher is a framework for performing logic-driven analysis on cross-chain data[1]. XChainWatcher leverages Soufflé [9], a state-of-the-art high-performance logical inference framework based on the Datalog programming language.

The workflow of XChainWatcher is presented in Figure 1. There are three phases: 1) decoding event and transaction data from blockchains, 2) building a set of logic relations based on the data extracted, and 3) evaluating relations using a set of detection rules. We design XChainWatcher to be generic and extensible, so that anyone can integrate support for any bridge. In addition, the logical rules can be fine-tuned for each supported bridge.

XChainWatcher leverages the concept of cross-chain model introduced in Hephaestus [3]. A cross-chain model captures the security properties of a bridge in terms of integrity, accountability, and availability [1]. Essentially, we model a set of anomalies through a set of cross-chain rules – which constitute a model – and compare it to real-world activity (fitness) to identify malicious behavior (attack) or unintended behavior. These rules are written in Datalog.

An example rule is `SC_ValidNativeTokenDeposit` that ensures a valid deposit of native tokens by the user in the source blockchain $\mathcal{S}$. This rule specifies a relationship between the transaction issued by the user, the event emitted by the bridge contract, and the event emitted by the contract representing the wrapped version of the native currency. In more detail, the checks are: **(1)** a bridge contract must emit a *Deposit* event; **(2)** there is a non-reverting transaction that transfers the same number of tokens natively in `tx.value`; **(3)** the token contract provided is indeed a version of the native currency of $\mathcal{S}$; **(4)** the validity of the token mappings (i.e., if users are trying to deposit tokens into the target blockchain $\mathcal{T}$ using a different token than what they are using in $\mathcal{S}$); and finally **(5)** the order of the events emitted by each contract (events emitted by token contracts precede events emitted

[1]https://anonymous.4open.science/r/XChainWatcher-B5F1/README.md

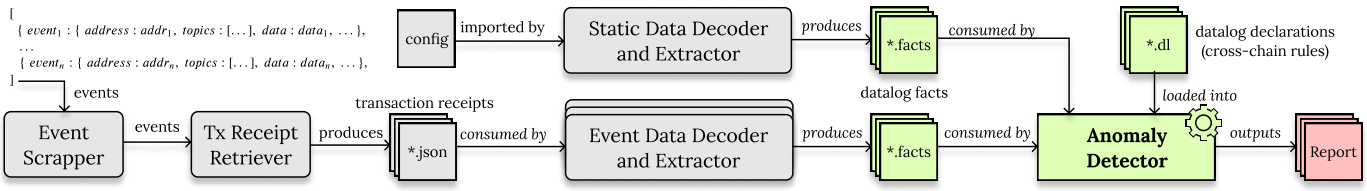

Fig. 1. XChainWatcher workflow.

by bridge contracts). In check **(2)** we do not check whether the transaction targets a bridge contract, as it may target an intermediary protocol contract (e.g., a bridge aggregator [12]), which in turn issues an internal transaction to the bridge. We only verify that the deposit event from the token contract must escrow tokens to a valid bridge contract, asserted using `bridge_controlled_address`. This rule ensures both that bridge contracts do not emit events asserting the deposit of tokens if the corresponding value was not effectively sent to the bridge, and the other way around. An attack that would be identified using this rule is the Feb. 2022 Meter.io hack [11].

```
SC_ValidNativeTokenDeposit(...args...) :-
  sc_token_deposited(tx_hash, bridge_evt_idx, _, _,
                dst_token, src_token, dst_chain_id, amount),
  sc_deposit(tx_hash, token_evt_idx, sender, bridge_address,
                                                    amount),
  transaction(_, src_chain_id, tx_hash, _, sender, _,
                                            amount, 1, _),
  token_mapping(src_chain_id, dst_chain_id, src_token,
                                                dst_token),
  wrapped_native_token(src_chain_id, src_token),
  bridge_controlled_address(src_chain_id, bridge_address),
  bridge_evt_idx > token_evt_idx.
```

III. Evaluation

We selected two previously exploited bridges to analyze the capabilities of XChainWatcher and the cross-chain rules: the Nomad bridge and the Ronin bridge. This selection allows us to test XChainWatcher against bridges that have suffered attacks and whose architecture and security assumptions differ. We used Blockdaemon's Universal API [5] to retrieve blockchain data from Ethereum mainnet, Moonbeam, and Ronin blockchains. We implemented a fallback to native RPC methods (namely *eth_getLogs* and *eth_getTransactionReceipt*) when the API could not provide the necessary data. We gathered addresses of interest, including proxies, and various versions of deployed contracts through documentation and analysis of the source code of each bridge.

We discovered significant attacks against cross-chain bridges: 1) transactions accepted in one chain before the finality time of the original one elapsed, breaking the safety of the bridge protocol; 2) users trying to exploit a protocol through the creation of fake versions of wrapped Ether to withdraw real Ether on the Ethereum blockchain; 3) bridge contract implementations handling unexpected inputs differently across chains hindering a good UX and leading to the loss of user funds. We found that although only 49 unique externally owned accounts exploited Nomad, there were 380 exploit events, with each address deploying multiple exploit contracts to obscure the flow of funds.

Acknowledgments. This work was financially supported by Project Blockchain.PT - Decentralize Portugal with Blockchain Agenda (Project no 51), WP6: Digital Assets Management, Call no 02/C05-i01.01/2022, funded by the Portuguese Recovery and Resilience Program (PPR), The Portuguese Republic and The European Union (EU) under the framework of Next Generation EU Program. This work was also supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020 (INESC-ID).

References

[1] A. Augusto, R. Belchior, M. Correia, A. Vasconcelos, L. Zhang, and T. Hardjono. Sok: Security and privacy of blockchain interoperability. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 234–234. IEEE Computer Society, may 2024.

[2] R. Belchior, L. Riley, T. Hardjono, A. Vasconcelos, and M. Correia. Do you need a distributed ledger technology interoperability solution? *Distrib. Ledger Technol.*, 2(1), Mar. 2023.

[3] R. Belchior, P. Somogyvari, J. Pfannschmidt, A. Vasconcelos, and M. Correia. Hephaestus: Modeling, analysis, and performance evaluation of cross-chain transactions. *IEEE Transactions on Reliability*, 73(2):1132–1146, 2024.

[4] R. Belchior, A. Vasconcelos, S. Guerreiro, and M. Correia. A survey on blockchain interoperability: Past, present, and future trends. *ACM Comput. Surv.*, 54(8), oct 2021.

[5] Blockdaemon. Blockdaemon rest api, 2024.

[6] J. Cirrone. $225 million raised in wormhole token sales, Nov. 2023.

[7] L. Duan, Y. Sun, W. Ni, W. Ding, J. Liu, and W. Wang. Attacks against cross-chain systems and defense approaches: A contemporary survey. *IEEE/CAA Journal of Automatica Sinica*, 10(8):1643–1663, 2023.

[8] T. Haugum, B. Hoff, M. Alsadi, and J. Li. Security and Privacy Challenges in Blockchain Interoperability - A Multivocal Literature Review. In *Proceedings of the International Conference on Evaluation and Assessment in Software Engineering 2022*, pages 347–356. ACM, June 2022.

[9] H. Jordan, B. Scholz, and P. Subotić. Soufflé: On synthesis of program analyzers. In *Computer Aided Verification: 28th International Conference, CAV 2016, Toronto, ON, Canada, July 17-23, 2016, Proceedings, Part II 28*, pages 422–430. Springer, 2016.

[10] Li.Fi. Li.fi – bridge & dex aggregation protocol, 2024.

[11] Rekt. Rekt - meter, 2022.

[12] S. Subramanian, A. Augusto, R. Belchior, A. Vasconcelos, and M. Correia. Benchmarking blockchain bridge aggregators. In *2024 IEEE International Conference on Blockchain (Blockchain)*, pages 37–45. IEEE Computer Society, aug 2024.

[13] R. Yin, Z. Yan, X. Liang, H. Xie, and Z. Wan. A survey on privacy preservation techniques for blockchain interoperability. *Journal of Systems Architecture*, page 102892, Apr 2023.

[14] M. Zhang, X. Zhang, Y. Zhang, and Z. Lin. Security of cross-chain bridges: Attack surfaces, defenses, and open problems. In *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*, RAID '24, page 298–316, New York, NY, USA, 2024. Association for Computing Machinery.

[15] Q. Zhao, Y. Wang, B. Yang, K. Shang, M. Sun, H. Wang, Z. Yang, and X. He. A comprehensive overview of security vulnerability penetration methods in blockchain cross-chain bridges. *Authorea (Authorea)*, Oct 2023.