Predicting the Effectiveness of Blockchain Bug Bounty Programs

[a]Ed Marcavage, [b]Jake Mason, [c,*]Chen Zhong
[abc]Skyes College of Business, The University of Tampa, Tampa, FL
[a]ed.marcavage11@gmail.com, [b]jakemason543@gmail.com, [c]czhong@ut.edu

Abstract

Bug bounty programs have proven to be an effective means for organizations to incentivize ethical hackers to report security vulnerabilities in their software. As the use of blockchain-based applications has grown, bug bounty programs have been established to identify vulnerabilities in these applications, such as smart contracts. However, bug bounty programs face unique challenges in encouraging ethical hackers. In this study, we collected data from about 200 bug bounty programs related to blockchain software from multiple bug bounty platforms. We analyzed the content of these programs and examined the involvement of ethical hackers, with the aim of examining the effectiveness of the current bug bounty programs for blockchain software. Additionally, we extracted various features from the content and format of the bug bounty programs and utilized them to construct a regression model that predicts the effectiveness of a program in drawing in ethical hackers. Our work is a fundamental step towards developing effective strategies for incentivizing ethical hackers in the blockchain domain.

Introduction

Bug bounties have proven to be an effective way of improving security and providing ethical hackers with an incentive. In the cybersecurity domain, bug bounty programs have played a crucial role in enabling companies to identify vulnerabilities in their software. As the use of blockchain-based applications has grown, bug bounty programs have been established to identify vulnerabilities in these applications, particularly smart contracts (Tang, Li et al. , Breidenbach, Daian et al. 2018).

Monetizing vulnerability explorations poses a major challenge for bug bounty programs in the blockchain domain. Unlike traditional bugs, where unethical hackers typically profit by holding data for a ransom or selling it on darknet marketplaces; there are few financial incentives to behave ethically when vulnerabilities are found in blockchain-based applications. When a hacker gains unauthorized access in the blockchain space, the rewards for reporting it are usually a small percentage of what it would be worth if they chose to exploit the bug and withdraw all of the money from the protocol.

This study aims to investigate the current state of bug bounty programs related to blockchain software as a preliminary step in increasing the effectiveness of bounty programs in attracting ethical hackers. To achieve this, we gathered approximately 200 bug bounty programs from various platforms, including BugCrowd, Hackenprood, ImmuneFi, and Hackerone. We also analyzed hacker records on Hackenproof to understand their contribution distribution. Through our analysis, we identified program features that could influence the effectiveness of bug bounty programs, including the length and wording of key sections and smart contract details such as the Solidity function types and if the source code was made publicly viewable. Additionally, we performed a preliminary regression analysis to examine how these metrics impact the program's effectiveness in attracting ethical hackers. The findings of this study have important implications for the community, as they highlight the challenges currently facing bug bounty programs and identify crucial factors that affect how well they attract ethical hackers. Our study aims to identify mechanisms that increase bug report submissions, ultimately improving the security of the cryptocurrency ecosystem and boosting confidence in the industry.

Related Work

Bug bounty programs have proven to be advantageous for companies that aim to engage ethical hackers in the process of discovering vulnerabilities (Maillart, Zhao et al. 2017). Research has found that the success of bug bounty programs is impacted by various features, including scoping, timing, submission quality, communication, and managing hacker motivation (Laszka, Zhao et al. 2018, Malladi and Subramanian 2019). It is important to provide clear reporting guidelines and incentives for fixing vulnerabilities, while maintaining transparency in reward ranges and eligibility (Malladi and Subramanian 2019). The structure of payouts is also a crucial factor to consider, as offering a flat fee may encourage hackers to prioritize

discovering numerous minor bugs over complex-critical vulnerabilities. (Maillart, Zhao et al. 2017).

The composition of the hacker community is also crucial to the success of bug bounty programs. Each security researcher brings a unique set of skills and perspective to the discovery process, uncovering bugs that others may miss (Maillart, Zhao et al. 2017). Hata, Guo et al. (2017) emphasizes the diversity of bug bounty program contributors and suggests that managers should differentiate between project-specific and non-specific contributors to maximize their impact. With the emerging of blockchain applications, bug bounty programs enhance cryptocurrency security, leveraging diverse environments and detecting more vulnerabilities.

Methodology

Data Collection

We have gathered comprehensive information on bug bounty programs. The collected information includes program descriptions, companies involved, covered categories (e.g., DEX, CEX, DeFi, NFT), target types (e.g., Dapp, smart contracts, web), and start/end dates. We also analyzed program documentation, submission guidelines, and reward types. Moreover, we examined the technology stack, report validation time, and reward distribution. Our data also include the number of participating hackers, submitted reports, and fixed bugs.

To ensure the consistency of the data format and minimize manual entry errors, we developed a Google Form for collecting bug bounty program data. The data sources for this study include various platforms, such as HackerOne, ImmuneFi, BugCrowd, and HackenProof. However, each platform has its unique approach to evaluating bug bounties, which means that some crucial information may be available on some platforms but not on others. Consequently, our dataset contains a significant amount of missing data for certain variables, such as the number of fixed bugs and the total number of reports submitted by hackers.

In addition to the information posted in the bug bounty programs, we further collected details of smart contracts using the Etherscan API by tracking the smart contract addresses when applicable. We examined the Solidity function type (e.g., payable, non-payable, pure, view) to determine the correlation between state-changing operations and the success of the bug bounty program. Vulnerability and verified ratios were calculated for each contract within a bounty program. The vulnerability ratio was calculated by pulling the Application Binary Interface (ABI) data of contracts using multiple API endpoints and collecting data from a total of 792 contracts across 48 bounty programs from 13 different blockchains. The verified ratio was calculated in a similar manner. A smart contract developer has the option to make their source code ‘verified’ (publicly viewable) or only the applications bytecode. The team assumed a higher vulnerability and verified ratio would increase hacker participation and the success of the bounty program.

Analysis

Our initial step involved data cleaning, which included removing duplicates and standardizing value formats. Next, for textual variables, we extracted features such as length, number of URLs, and any provided smart contract addresses. We performed an exploratory analysis of the data by examining the outliers, checking the distribution of the variables, and exploring their correlation. To avoid multicollinearity in the regression analysis, highly correlated variables were removed from the dataset.

Following the data processing and analysis, we identified 14 independent variables for our analysis. The dependent variable, representing the success of the bounty program, was calculated as an average based on four features, the total number of: hackers, bugs fixed, reports submitted, and rewards paid out. Despite encountering outliers and missing values in our dataset, we applied an OLS regression model to fit our data and performed 10-fold cross-validation to obtain a more robust estimate of model performance.

Preliminary Results and Future Work

The results of our correlation analysis found that there is a positive correlation between the number of hackers involved and the number of bugs identified. This finding suggests that the participation of more ethical hackers is associated with the detection of more bugs. Additionally, we observed that bug bounties with clear guidelines, and comprehensive documentation were easier for hackers to navigate and as a result, attracted more hackers. The analysis of hackers on the HackenProof platform revealed that 1% of the hackers accounted for 42.5% of the total number of bugs submitted on the platform. The regression analysis results suggested that the independent variables used in the model explain approximately 65% of the variance in the program success. Among the independent variables, the age of the program, the number of URLs, and the total length of the program were found to be statistically significant and have a positive impact on the dependent variable.

In the future, we plan to expand our dataset by collecting more bug bounty programs in the blockchain domain and enhancing our regression model with a larger and more diverse dataset. Additionally, we have observed the emergence of novel bug bounty programs that have bounty rewards built into their blockchains, and we aim to compare the effectiveness of various types of programs in terms of attracting and engaging hackers.

References

Breidenbach, L., P. Daian, F. Tramèr and A. Juels (2018). Enter the hydra: Towards principled bug bounties and exploit-resistant smart contracts. 27th {USENIX} Security Symposium ({USENIX} Security 18).

Hata, H., M. Guo and M. A. Babar (2017). Understanding the heterogeneity of contributors in bug bounty programs. 2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE.

Laszka, A., M. Zhao, A. Malbari and J. Grossklags (2018). The rules of engagement for bug bounty programs. Financial Cryptography and Data Security: 22nd International Conference, FC 2018, Nieuwpoort, Curaçao, February 26–March 2, 2018, Revised Selected Papers 22, Springer.

Maillart, T., M. Zhao, J. Grossklags and J. Chuang (2017). "Given enough eyeballs, all bugs are shallow? Revisiting Eric Raymond with bug bounty programs." Journal of Cybersecurity **3**(2): 81-90.

Malladi, S. S. and H. C. Subramanian (2019). "Bug bounty programs for cybersecurity: Practices, issues, and recommendations." IEEE Software **37**(1): 31-39.

Tang, Y., K. Li, Y. Wang and J. Chen "Ethical Challenges in Blockchain Network Measurement Research."