УДК 658.52.011.56(06)

Ястремской А.А.,
студент 2 курса магистратуры по направлению
«Информационная безопасность»
Севастопольский государственный университет
Россия, Севастополь
Браславская Е.А.,
научный руководитель, кандидат педагогических наук,
доцент кафедры иностранных языков
Севастопольский государственный университет

ОПТИМИЗАЦИЯ ОСНОВНЫХ ПРОБЛЕМ В ПРОГРАММАХ ВОЗНАГРАЖДЕНИЯ ЗА ОШИБКИ

Аннотация: *В данной статье рассматриваются системные проблемы, присущие современным платформам управления программами Bug Bounty. Проводится анализ ключевых вызовов, с которыми сталкиваются исследователи безопасности: коммуникационные разрывы, спорная модерация, нечеткость scope и дисбаланс в оценке вознаграждений. На основе эмпирических данных и анализа существующих практик предлагаются стратегии протоколизированного взаимодействия, направленные на оптимизацию процесса для обеих сторон – исследователей и владельцев программ. Доказано, что устойчивость экосистемы багбаунти зависит от решения этих институциональных, а не только технических, проблем.*

Ключевые слова: *Bug Bounty, кибербезопасность, управление уязвимостями, платформы багбаунти, человеческий фактор.*

Yastremskoy A.A.,
2nd year Master Student in Information Security
Sevastopol State University
Russia, Sevastopol
Braslavskaya E.A.,
scientific Advisor, Candidate of Pedagogical Sciences,
Associate Professor of the Foreign Languages Department
Sevastopol State University

OPTIMIZATION OF MAJOR ISSUES IN BUG BOUNTY PROGRAMS

Abstract: *This paper examines the systemic challenges inherent in modern Bug Bounty Program management platforms. Key challenges faced by security researchers are analyzed: communication gaps, contentious moderation, scope ambiguity, and imbalances in reward evaluation. Based on empirical evidence and analysis of existing practices, protocolized communication strategies are proposed to optimize the process for both parties - researchers and program owners. The article argues that the sustainability of the bugbaunti ecosystem depends on addressing these institutional, not just technical, challenges.*

Keywords: *Bug Bounty, cybersecurity, vulnerability management, bug bounty platforms, human error.*

Introduction

The crowdsourcing model of security, implemented through bug bounty platforms (such as HackerOne, Bugcrowd, Intigriti), has proven to be highly effective for identifying vulnerabilities. However, the focus of the community has traditionally been on the technical aspects of finding vulnerabilities, while the socio-technical and institutional challenges of the platforms themselves remain in the shadows. These challenges act as significant barriers to productive

collaboration between researchers and organizations, leading to demotivation of professionals, lower quality reports, and ultimately a decrease in the overall effectiveness of the model.

This article aims to systematize the main classes of Bug Bounty Program (BBP) systemic problems and to develop structured strategies for researchers to overcome them, based on the principles of professional communication, documentation and process approach.

Security of information systems is a key aspect of modern society, because information systems play an important role in our lives and are used in various spheres. Security is critical for protecting confidentiality, integrity and availability of data, therefore it becomes an object of interest for attackers, and the number of security threats to information systems is only increasing. [1].

The programs work according to this principle: participants around the world register on special platforms, learn the rules and start checking systems. After completing the task, they pass the bugs found to the organizations and sometimes help to fix them.

If bugbounty is an opportunity for researchers to pump up their professional skills and earn money, then for companies it is an opportunity to find and fix vulnerabilities. They don't need to constantly hire expensive pentesters and testers. It is easier to pay a certain amount of money for a bug found than to spend months searching for candidates. In addition, bugbounty can involve both students and highly qualified specialists, the staff of which not every company can afford.

Gradually, the idea of involving third-party researchers was fully developed in the IT community. Companies were no longer so skeptical, so they began to launch their own bugbounty programs.

One of the most famous Zero Day Initiative programs was launched by Tipping Point. It paid for vulnerabilities with high impact and PoC (proof-of-concept, demo versions of exploits). Technology giants also paid attention to this practice. Google adopted it in 2010, Microsoft in 2013, and then other

corporations. Around the same time, the largest Russian companies also started using bug bounties: Yandex in 2012 and VK in 2013. Large, fast-moving companies have already adopted the use of Bug Bounty in conjunction with Pentest.using Bug Bounty in conjunction with Pentest. This allows them toto examine vulnerabilities in an information system from multiple angles, reduce the amount of timeto add testers to their testing teams and save money. Suchofferings can be found on aggregator sites, the most well-known of which are.which are: HackerOne, Bugcrowd, Vulnerability Lab, BountyFactory,Synack. [2].

Startups HackerOne, Bugcrowd, and Synack offered companies ready-made sites for launching bugbounties. They took care of vetting researchers, processing reports and payments, turning the process into a convenient service. Now, organizations don't need to build their own infrastructure – all they had to do is to publish the rules and budget.

To prevent researchers from going too far, companies began to form what is known as a scope – a list of systems and methods by which a company can be "broken". Going outside the scope can still be considered an intrusion and will definitely not be paid for.

Consequently, despite its impressive growth and institutionalization, the very complexity of this distributed model creates "friction" at the points of interaction between researchers, platforms and companies. This gap between the technical potential of crowdsourcing and its operational realization represents the next major obstacle to the development of the entire ecosystem.

Identified problems in BugBounty

The traditional result that companies participating in bugbounty get is the identification of vulnerabilities in their information systems. Unlike a classic pentest or application audit, in this case money is paid for bugs found, not for the time spent by researchers on their work Going beyond the direct relationship between a program and white hats, the rules likely also shape the competitive process between the different programs on a platform. The stated terms may

contribute to attract white hat researchers, or they may dissuade them from working for a particular program. All these factors are so far largely unexplored [3].

Thus, company can face with the following problems:

1. Messaging flow

Popular Bug Bounty programs can generate hundreds of messages per month. If you don't have the processes in place to handle them, you'll quickly get swamped with duplicates, false positives, and spam. Researchers will be frustrated with slow responses and your team will be frustrated with constant distractions. This is especially painful for companies new to security. Imagine that you decided to study cooking and on the first day you invited a dozen professional chefs home to criticize your dishes. The stress is guaranteed.

2. Shortage of money to fix the problems found

Finding a vulnerability is only half the battle. It needs to be fixed, the patch needs to be tested, and an update needs to be released. If you don't have the development resources to fix bugs quickly, Bug Bounty will become a source of accumulating technical debt. Researchers may start publicizing uncorrected vulnerabilities, creating additional reputational and security risks.

3. Unreadiness of the product

Running a Bug Bounty program for a product in active development is a bad idea. Researchers will find vulnerabilities in the code you plan to rewrite next week. It's better to wait until the architecture and core functionality are relatively stable. This will also avoid situations where

4. Basic security issues

If your application still uses outdated libraries, doesn't validate user input, and stores passwords in plaintext, Bug Bounty will become an expensive way to find out what you should already know. Such obvious problems are better found and fixed with automated tools or basic auditing than paying researchers to find them.

5. Negative publicity

Some researchers may publicize the vulnerabilities they find, especially if the company is slow to respond or refuses to fix the problems. If your company is not ready for such publicity, it is better to use private programs or limit pentests altogether. This is especially important for companies in conservative industries or those just starting to build a reputation for security.

Having analyzed a number of problems the following assessments of their criticality were given and reflected in the bracket diagram:

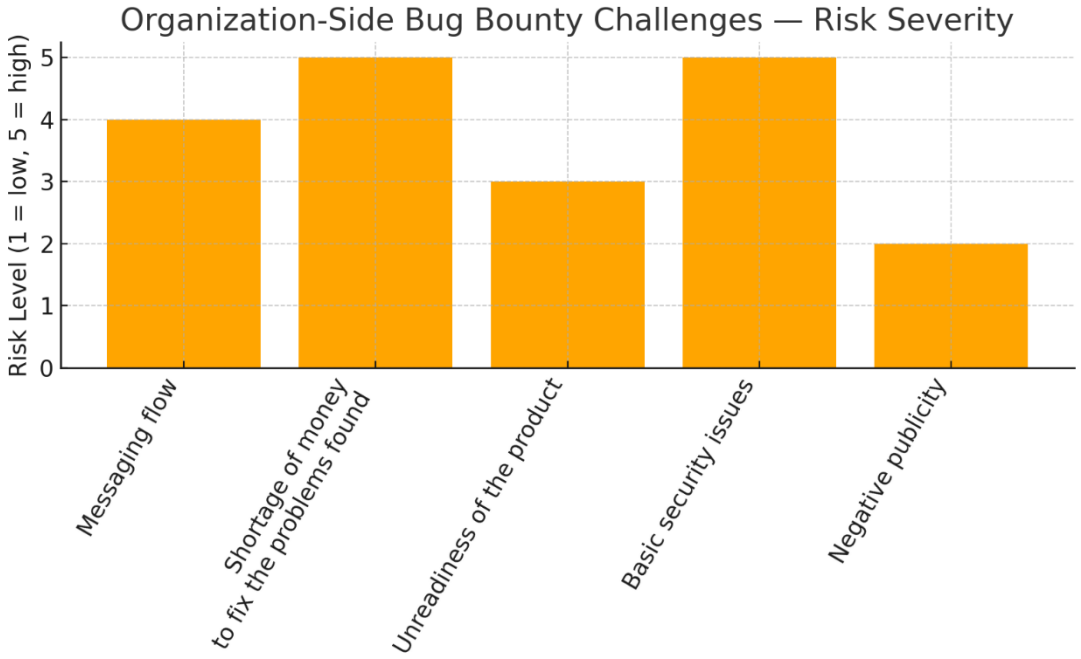

Figure 1. Evaluation of key issues in Bug Bounty programs

To assess risk, we used a composite model inspired by standards for cyber risk assessment and security program maturity. The approach is based on the principles of: ISO/IEC 27005, FAIR Framework, CVSS v3.1, Bug Bounty Readiness Guidelines.

The estimate was calculated as follows:

Metrics:

- **L – Likelihood** (probability / how often it happens)
- **I – Impact** (effect on business / security)

- **P – Prevalence** (scale / how many users or components are affected)
- **R – Relevance/Timeliness** (current importance / how relevant it is now)
- **V – Visibility/Publicity** (risk of exposure / public attention)

$$L = 0.28,\ I = 0.32,\ P = 0.18,\ R = 0.15,\ V = 0.07$$

The result is then normalized to a 0-100 scale for a visual graph.

Although the profession of bughunter looks like a dream job, with the growing popularity of bugbounty programs come challenges. Let's look at the main ones.

Accessibility of entry into the industry is both the boon and bane of bugbaunti. Participants only need to register on the platform and start looking for vulnerabilities; successful findings are rewarded. Bug bounty programs attract a huge number of newcomers, but some of them approach the process irresponsibly and send reports with superficial findings or “low-quality vulnerabilities” – for example, the lack of CSP headers on the test page.

Such reports create information noise, complicating the work of both companies and platform moderators. Platforms usually have a rating of the best bughunters, where not only the number of vulnerabilities is taken into account, but also their criticality. Also, a large list is published on the program page at once: which vulnerabilities are accepted and which are not.

But, on the other hand, companies do not always behave ethically. Some ignore researchers' reports for months, don't give feedback, don't update the vulnerability status – and then six months later suddenly close the report with the words, “Thank you, but this is a duplicate. We already knew that”. There is no reward, of course. Bug bounty is a practice where companies offer researchers rewards for finding vulnerabilities in their products and IT infrastructure [4].

On platforms like Standoff 365 or BI.ZONE BugBounty there are SLAs (Service Level Agreements) – clear commitments on response time and report processing. For example, a company commits to acknowledge receipt of a vulnerability within five business days and close it depending on the criticality - from 30 days to six months. This reduces the number of “hung” reports, gives researchers an idea of what to expect, and affects the company's reputation.

Enhancing the Sustainability and Efficacy of Bug Bounty Programs

The paradox of a successful Bug Bounty program is that its effectiveness directly generates an operational load that can paralyze a security team. The initial flood of reports, characterized by a high percentage of duplicates, false positives, and spam, is not the result of researcher bad faith, but a natural state of unfiltered crowdsourced signal.

At this level, the system automatically filters out obviously incorrect reports using a set of rules (rule-based engine) and ML models. In addition, bugbounty can involve both students and highly qualified specialists, the staff of which not every company can afford.

To handle this operational pressure efficiently, modern bug bounty platforms implement several automated mechanisms that ensure only relevant and actionable reports reach human moderators:

1. Scope validation: Automatically checks whether the target asset belongs to the In-Scope list. Reports targeting Out-of-Scope assets are rejected automatically with instant notification and reference to program policy.

2. Duplicate Check: The system automatically compares a new report against a database of previously accepted reports using hash sums of key parameters (endpoint, vulnerability type, attack vector). Highly likely duplicates are flagged for instant rejection by a moderator.

3. Pattern Analysis: Using simple rules to weed out common spam and false positives (e.g., reports referencing vulnerabilities in third-party CDNs, no proof of exploitation).

4. Result: 30-50% reduction in human resources by automatically rejecting incorrect noise.

The following diagram (Figure 2) illustrates the tiered triage and automation framework designed to manage high volumes of incoming bug reports efficiently.

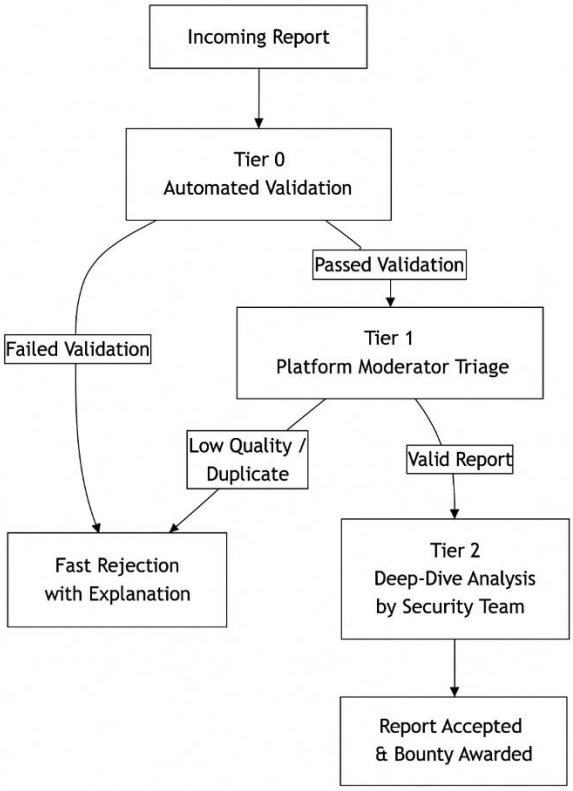

Figure 2. Proposed Multi-Level Report Processing Framework

Conclusion

The conducted research has achieved its goal – a comprehensive study and systematization of possible measures to prevent key problems in the bug bounty ecosystem. In the course of the work, all the objectives were consistently solved.

First, we identified and studied in detail the challenges faced by all participants of the ecosystem: researchers, security teams, and program organizers. The challenges were found to be organizational and communication rather than technical, including operational overload, communication lags, contentious moderation, and imbalances in the rewards system.

Second, the analysis identified a range of the most pressing challenges that have the greatest negative impact on program effectiveness. These include: unpreparedness of organizations to handle a large flow of reports, leading to operational collapse; lack of resources to fix the vulnerabilities found, turning the program into a source of risk; and systemic friction on the researcher side, such as lack of feedback and transparency in the moderation process.

Third, the work identified and theorized solutions for the identified problems.

References:

1. Тутов, В. А. BAS – Pentest – RedTeam – BugBounty / В. А. Тутов // Методика преподавания математических и естественно-научных дисциплин: современные проблемы и тенденции развития : материалы XI Всероссийской научно-практической конференции, Омск, 27 июня 2024 года. – Омск: Омский государственный университет им. Ф.М. Достоевского, 2024. – С. 189-195.

2. Раздымаха, И. А. Роль программы BUG BOUNTY в информационной безопасности / И. А. Раздымаха, Н. С. Афанасьева // Инновационные проекты и технологии в образовании, промышленности и на транспорте : Материалы девятнадцатой научной конференции, посвященной Дню Российской науки, Омск, 06–11 февраля 2025 года. – Омск: Омский государственный университет путей сообщения, 2025. – С. 389-395.

3. Li, V. Bug Bounty Bootcamp: The Guide to Finding and Reporting Web Vulnerabilities. – San Francisco: No Starch Press, 2021. – 319 p.

4. Laszka, A. The Rules of Engagement for Bug Bounty Programs / A. Laszka, M. Zhao, A. Malbari, J. Grossklags // Financial Cryptography and Data Security: 22nd International Conference, FC 2018. – Heidelberg: Springer-Verlag, 2018. – P.138-159.