Softwares often have security vulnerabilities and can be attacked by adversaries, with potentially significant negative economic consequences. This is particularly critical for blockchain infrastructure providers. Once the software is deployed, there is no turning back or hardly any legal defense mechanisms against system exploitation.Therefore, such projects rely on public intrusion test where everyone was allowed to probe the software and report any vulnerabilities (bugs) in exchange for monetary rewards (bounty). This type of program, often called bug bounty or crowdsourced security, has become a major tool for detecting software vulnerability on blockchains through publicly announced bug bounties. Bug bounty programs are also used by governments and tech companies.

In this project, we offer insights on some of the dimensions of bug bounty design, using a game-theoretic model of a simple contest, where agents with different abilities decide on whether or not to exert costly effort for finding bugs. We focus on the simple problem how to maximize the likelihood to find bugs when a given amount of money is available for rewards. In particular, we will focus on three design variables for bug bounty systems. How large should the crowd of agents invited to find bugs be? Should paid experts be added to the crowd of invited bug finders? Should artificial bugs be added to the software to increase participation in bug finding and to increase the likelihood that the real bug is found?

To answer these questions and other, general questions about the nature of equilibria in bug bounty schemes, we develop a simple model of crowd-sourced security. A group of individuals of arbitrary size is invited to search for a bug. Whether a bug exists is uncertain. The individuals differ with regard to their abilities to find bugs, which we capture by different costs to achieve a certain probability to find the bug if it exists. Costs are private information. The designer of the bug bounty scheme offers a prize for the individual or the set of individuals who find the bug. The designer can vary the size of the group of individuals invited to find a bug, can add a paid expert to the crowd, and can insert an artificial bug with some probability.

We obtain the following results. First, we establish that any equilibrium strategy must be a threshold strategy, i.e. only agents with a cost of search below some (potentially individual) threshold participate in the bug bounty scheme. Second, we provide sufficient conditions for the equilibrium to be unique and symmetric. Third, we show that even inviting an unlimited crowd does not guarantee that bugs are found, unless there are agents which have zero costs, or equivalently have intrinsic gains from participating in the scheme. It may even happen that having more agents in the pool of potential participants may lower the probability of finding a bug. Fourth, adding paid agents can increase the efficiency of the bug bounty scheme, although the crowd that is attracted becomes smaller. Fifth, we illustrate how adding (known) bugs is another way to increase the likelihood that unknown bugs are found. When the additional costs of paying rewards are taken into account, it can be optimal to insert a known bug only with some probability. Finally, we illustrate the equilibria numerically when costs are distributed uniformly and identify circumstances when asymmetric equilibria occur.

See https://arxiv.org/pdf/2304.00077.pdf for more details.