Simulated Fraud Squad

Fraud Detection and Defense is the closest workstream that this squad could work under, however it's more of a preventative squad as opposed to a defensive one.

Definitions:

QF (Quadratic Funding/Quadratic Finance)

If you are unfamiliar with quadratic funding/finance, check out this website. For more rigor, you can also read the whitepaper. Another great resource is Vitalik's primer.

Human Verification

Human Verification is a problem addressed by projects like Proof of Humanity, BrightID, POAP, and others. Moving forward we will use the **HV** acronym.

Direct Fraud

There are many types of fraud for the matching pool, for this squad's purposes we are exploring the possible types that are **directly impacted by the dynamics of the grants system**. The **bolded parts** illustrate the differences between the types:

- Type 1 (sybil)
    - **Exploitation of QF** by splitting larger contributions into smaller ones to get more funding from the pool into one's own grant by **employing botnets (sybil accounts)**.
    - Exploiter has $N, splits into Nx$1 contributions, gets out $N+matching funds **without external cooperation** (fighting the optimality gap).
    - **Have to worry** about gas fees for funds splitting.
    - **Has to worry** about the amount spent on donations to "good grants" in attempt to obfuscate the contribution graph
- Type 2 (collusion)
    - **Exploitation of individuals participating in QF** by splitting larger contributions into smaller ones to get more funding from the pool into one's own grant by **employing other humans/human's accounts**.
    - Exploiter has $N, splits into Nx$1 contributions, gets out $N+matching funds **with external cooperation** (fighting the optimality gap).
    - **Don't have to worry** about gas fees for funds splitting.
    - Also includes purchasing other account's private keys.

Both direct fraud types have a lot of overlap, but some key differences. Mainly, Type1 is purely sybil accounts and Type2 is purely human accounts. An attacker may use a blend of Type1/Type2 but they are still distinct concepts and are useful to address separately.

Indirect Fraud

Both types of direct fraud also have the ability to cause **indirect** stealing from the matching pool. By increasing the amount of donations a grant has, you increase the likelihood for other non-colluded users to donate to said grant. This is **market manipulation**.

Solving Type1 Fraud (Direct/Sybil)

This squad's **initial** focus is to solve type1 fraud.

Hypothesis:

We can stop **type1** fraudsters stealing from the grant matching pool by modifying the QF funding mechanism instead of complex fraud detection methods (like machine learning).

Proposed Solution:

Develop a simulation of the gitcoin grants system (or extend block science's cadcad simulation) and compare modifications of QF with respect to the optimality gap.

One possible modification of QF would be to add a taxation to each individual contribution (taxes go straight to the next matching pool). We call it "Residual Quadratic Funding" or "ResQF" (naming inspired by the ResNet architecture). This taxation would be high for small contributions and small for high contributions (to de-incentivize sybil splitting). This method would rely on less assumptions to be made than the current FDD-sybil efforts, and if true would be very robust.

QF

$$\{F^p\}_{p\in P} = \Phi^{QF}(c_i^p) = \left\{(\sum_i \sqrt{c_i^p})^2\right\}_{p\in P}$$

source

ResQF

$$\{F^p\}_{p\in P} = \Phi^{ResQF}(c_i^p) = \left\{(\sum_i \sqrt{c_i^p\, \alpha(c_i^p)})^2\right\}_{p\in P} ,$$

where the $\alpha(c_i^p)$ *function is smooth, and increasing OR constant with a range of* $[0,\ 1)$

Then the sum of all of the taxes: $\sum_p(\sum_i\left[c_i^p(1 - \alpha(c_i^p)\right])$ is added to the next matching pool.

Another idea is to use a human verification as a parameter to the α function. That way, we can for example tax users with more strong signals of being a human less than those without.

Example

Say you have $1000 and want to donate to your grant so you can maximize the grant matching (by splitting).

If you donate without fraud, that means you would just send 1x$1000 transaction (costing $1000 total, assuming 0$ gas fees).

Normal QF Fraud

You can fraudulently donate 1000x$1 (costing $1000 total, assuming 0$ gas fees). This means the grant you donate to will receive a much larger portion of the matching pool than it was supposed to.

ResQF Fraud

Let’s see how this plays out using ResQF with **hard-coded values**;

- α($1) = 0.8 (20% tax)
- α($1000) = 0.99 (1% tax)

Note: the α function values defined above are not real values, the real ones will have to be determined in a simulation. We might even want to have this function be normalized against all total transactions received in the round and/or maybe even respond to gas prices.

You can fraudulently donate 1000x$1 still, however every $1 contribution will be taxed at 20%. This means the grant will receive $800 instead of $1000. The other $200 will go to the next grant round’s matching pool.

On the other hand, if you donated the $1000 in a single transaction, the grant you’re donating to would receive $990.

Implications

Taxing shrinks the optimality gap. As the $1 tax increases, it becomes increasingly risky to commit fraud (in the same way that gas price increasing does). The required capital for fraudsters grows with the tax rate, and if they don’t perfectly hit the thin optimality gap range, all their profits would go directly to the community next round.

When a fraud attacker is punished in this way, the future community receives a proportional reward. Kind of like a weighted (and automatic) class action lawsuit.

Solving Type2 Fraud (Direct/Collusion)

This squad's **future** focus is to solve type 2 fraud.

Hypothesis:

We can stop **type 2** fraudsters (colluders) stealing from the grant matching pool by:

- High quality human verification. This helps solve the sybil problem by connecting one account with one human. This account must be directly connected to their proof of personhood. E.g. a new privacy preserving account is generated and can be funded from other non-private sources.
- Money can only be distributed to grants from this account in an anonymous way. This makes it hard for main colluders to know that the people they are paying off (or intimidating) are donated to the correct grants.
- The funds only get distributed at the end of a round. This is important as it gives people time to redistribute funds if needed (further discussed below).
- The main colluders can still just ask the network of people to hand over the private keys to their private accounts. Therefore it is important that each person can, in a completely anonymous way, redistribute funds from this private account by verifying their identity. The redistribution must only be possible once per proof of personhood verification. This prevents the main colluders from just spamming the same distribution. Secondly, the private keys that the main colluders have must not be able to see redistributions and only the distribution they specified.
- This creates doubt in the minds of colluders. They can never be certain that the people they are funding are not just redistributing funds instead of listening to them. One can then even ask honest users to take these birbes and just distribute funds as they normally would. The colluders then just lose resources as time goes on.

Proposed Solution:

The future work of this squad is to propose a solution that suffices our hypothesis' requirements. However, we have some suspicions that the solution **may** involve:

- A new non-ETH blockchain to be developed.
- Collaboration with one or many HV projects.
- Standardized intra/inter-DAO task management (for HV).

Addressing Market Manipulation (Indirect Fraud)

TODO – should our squad even be responsible for this? Can we even solve market manipulation in a decentralized way?

One idea: If we go the fully anonymous route (for solving type2 fraud), then we may also want to keep grant contributions fully anonymous until the grant round is over entirely to prevent this. It is unclear how possible this would be.

Roster/Funding/Operations

For this project and the current knowledge we have, it seems that a squad of size 2-4 members is ideal. This number may increase or decrease as time goes on.

A thorough understanding of the surrounding literature is important to finding a simple solution and evaluating it properly. That means the amount of research work required for this squad is high, so tangible results won't be immediately observable.

Current Roster

This roster is subject to change, we are actively pursuing new members.

- [lead] Dyllan (nollied#6773)
- Dries (Dries#0479)
- Chris (steegecs#2390) [Not yet onboarded into FDD!]
- Kylin (Kylin#5100)

Major Operations

- Collaborative research.
- High bandwidth async/sync collaborative ideation.
- Simulation development and evaluation of realism. A low quality simulation means low quality results. This simulation should be modular and simple.
- Funding mechanism modification development and evaluation of realism/any other implications.
- Simple presentation of results.
- If the results are solid and permission is granted, implementing the modifications into Gitcoin's actual grant funding mechanism.

Initial Weekly Funding Request

$650 per member per week = $650 x 4 = $2,600/week.

Work for this project began **12/13/2021** with our thread in the fdd-sybil channel "Sybil Assumptions Discussion" and private correspondence/research/ideation between nollied & dries.

Funding Justification

- The amount of research requires us to spend a lot of time with each resource (and being 100% synchronized) before moving forward to the next. Also reaching out to resource creators for advice/more understanding (which we've already done).
- The FDD specialized knowledge required for this project is very high and it is hard to onboard fresh eyes that don't already have experience in the FDD workstream/sybil squad.
- A background across the fields of statistics, game theory, reinforcement learning, machine learning, software engineering, economics, blockchain development/cryptography, etc. is essential for all members.