Contest Proposal: SMV Smart Contract System Phase 1 Formal Verification

Short Description

The contestants shall perform the Phase 1 formal verification of the central SMV smart contracts (Demiurge, Padavan, and Proposal from https://github.com/RSquad/BFTG (commit hash - 013f691a0a603a5a0a9dd99e3b6558a5cc54e8a3)), hereinafter referred to as “smart contracts”, including the preformal audit of the same contracts. All debot contracts are excluded from the present contest.

Motivation

The contest shall yield a set of specifications necessary for the security audit and formal verification of the smart contracts. These shall include a function-level specification in a formalized version of English, a business-level specification, a description of user scenarios, and an evaluation of smart contract security and reliability threats (further details as to the specific content of said specifications are provided under **Contest Terms**). All these activities are critical as the preliminary steps for getting (at the latter phases) the formal specification and, eventually, the formal verification of the smart contract.

As a supplementary step to the development of the above-mentioned specifications, it is hereby proposed that the contestants also undertake a **preformal audit** of the smart contracts. The rationale of the preformal audit is to reduce the workload at the stage of formal verification proper (since such audit can help identify and remove bugs that would otherwise surface only in the course of the formal verification, necessitating major rework) and to speed up the release of the smart contracts to Free TON, pending complete formal verification.

Caveat: Given the risks associated with possibly overlooked vulnerabilities, the release of smart contracts for which the full formal verification has not been completed should be undertaken exclusively in case of a strong business need.

Prerequisites

Before entering the preformal audit process, each smart contract must comply with the following rules (otherwise, the contest may be frozen by the request of any participant with the corresponding end date shifting until these requirements are met):

- The smart contract must be compilable with the latest version of the Free TON development toolchain
- All smart contracts must be submitted with detailed documentation

- The smart contract's developers must set up a Telegram group for discussions as well as a slot for one-hour voice calls at least once a week
- All the smart contracts being audited must be covered by unit and integration tests
- For the unit tests, the coverage must be measured (by any kind of coverage tool) and provide an evidence to the numbers not lower than:
    - Public function (all the functions but *private*) coverage - 100%
    - Private function coverage - 95%, all the functions having more than 5 lines of code must be covered
    - Line coverage - at least 80%
    - All the tests must pass
- For the rest of the tests, all the tests must pass

The contestants shall use all information obtainable from the smart contract system developers (including, but not limited to, specifications, interviews with developers, and (prototype) source code).

Contest Terms

The contestant shall submit a document in the PDF format that can be used by anyone during the latter phases that will include the following:

1. General
    a. A general description of the smart contracts to be verified, accompanied by a chart
    b. A list of roles and responsibilities that can be identified for the smart contracts of the system
    c. A description of the semi-formalized variant of English that the contestant intends to use for smart contract specification
    d. A description of the preformal audit of the smart contracts

2. Smart contract specification
   (Note: there shall be a subsection for each of the smart contracts in the system; the structure of all such subsections is outlined below)
    a. A description of the smart contract state, including state components and their description.
    b. A function-level specification in the above-mentioned semi-formalized variant of English that shall describe, for each function:
        i. Access requirements
        ii. Input parameters
        iii. Output parameters
        iv. Exceptions
    c. A business-level specification that shall:
        i. Be written in plain English

        ii. Contain a set of common-sense logical statements
        iii. Be accompanied by diagrams and flowcharts
        iv. Include role-action matrices

3. User Scenarios
   This part shall contain the description of the user scenarios identified for the smart contract system

4. Security and Reliability Evaluation
   This section shall contain a list of identified security and reliability threats. For each list item, the threat's severity evaluation and the ways of mitigating it shall be described.

5. Preformal Audit Report (this is not a mandatory part, however additional scores are provided in case of completion)

   This section, supplementing the above-mentioned specifications, shall include the results of the preformal audit of the smart contracts undertaken by the contestant, complete with a list of identified bugs and/or vulnerabilities discovered in the course of such audit. In particular, this section shall contain the following information:
   a. A high-level description of the contestant's approach to preformal audit
   b. A description of the toolkit(s) the contestant intends to use for preformal audit purposes (if any), including their motivation to do so
   c. A list of the checkpoints the contestant will make before commencing preformal audit both for functional and application level where application level checklist must include:
      i. Correctness checklist (correct state always moves to correct)
      ii. Liveliness checklist (no stall in any conditions)
      iii. Safety checklist (any foreseeable attacks are checked)
   d. A list of potential holes the contestant plans to leave opened accomplished with the solutions intended to reduce the risk of malfunction
   e. Results of the applying checklists and well as the tools (if any)
   f. Result summary

6. Contestant Information
   This section shall contain the information about the contestant. The contact information (preferably a Telegram ID) and a short overview of the contestant's background and experience with blockchain, security, and formal verification are obligatory. Otherwise, the contestant is free to include any further information as they see fit.

Contest Dates

27 April 2021 - 17 May 2021

Proposed Prices

The total contest budget is **200 kTON**, whereby 95% are allocated to the contestant awards and 5% are allocated to the jury reward.

The contestant awards are distributed as follows:

- Place 1 - 100 kTON
- Place 2 - 60 kTON
- Place 3 - 30 kTON

The Jury

The Jury shall be formed from acknowledged experts in the fields of security, smart contract audit, and formal verification fields, whereby:

- Jurors whose team(s) intend to provide submissions in this contest shall lose their right to vote in this contest
- Each juror shall vote by rating each submission on a scale of 0 to 10 (0 equalling rejection of the proposal); a juror may abstain from voting if they do not see themselves sufficiently qualified to judge such proposal
- A juror that has voted on a submission shall provide detailed feedback on it

Jury Guidelines

- The main goal of the jury is to check how the provided specification is accurate and full.
- The specification is intended to be understandable for an average IT professional but at the same time it must be evident it's relatively easy to convert it to the formal one
- All the requirements mentioned above are considered as mandatory otherwise some points have to be taken from the corresponding application. The only exception is the preformal audit that is optional but brings some additional points
- For the preformal audit the main goal of the jury is to check how the performed audit is useful in terms of decreasing the bug risk for the contract being audited
- Any team that managed to find a non-minor bug must be rewarded with some additional points
- Any bugs related to the kind of the exceptions as well those related to the logging or retrieving the state are considered as minor and should not be taken into account for the ranking

Jury Rewards

The total budget for jury rewards is 5% of the above-mentioned contest reward budget (
10 kTON).

This amount shall be distributed between jurors who have voted **and** provided feedback on submissions.

The proportion of the total budget assigned to a juror shall be defined according to the extent of this juror's participation in the contest, i.e. the count of votes cast by this juror divided by the total votes cast count for this contest.

Procedural Limitations

- Only one submission per contestant shall be accepted. Multiple submissions, including but not limited to updated versions of the initial submission, are not allowed.
- Submissions shall be made within the time frame defined above in the **Contest Dates** section. Late submissions shall be rejected by the Jury.
- All submissions shall contain the contestant's contact information (preferably a Telegram ID) to ensure that the jury can match the submission to the specific contestant. If such contact information is missing, the submission shall be rejected.
- If the submission contains links to external material (reports on further work by the contestant), this material shall have the contestant's contact details (preferably a Telegram ID), to ensure that the jury can match the material to the specific contestant. If such contact information is missing, the submission shall be rejected.

Disclaimer

Anyone can participate, but Free TON cannot distribute tokens to US citizens or US entities.