Security assessment preparation checklist

Smart contract assessment on Tezos

Version / Date	Description
1.1 / 01.12.2021	Version 1.1
1.2 / 24.12.2021	Example added.
1.3 / 19.10.2022	Testnet deployment and on-chain testing.

Purpose

This document defines the requirements for a project to be ready for a security assessment of its smart contracts on the Tezos Blockchain. These requirements are set out to foster an efficient security assessment.

Documentation

Item	
General description of the smart contract about its purpose/intention and a description of what the use case will be resp. how the smart contract will be used.	
For each entrypoint and smart contract internal function the behaviour is specified. This includes at least: • Expected input parameters (including type of parameter and whether any precision factor is applied) • Expected result • Access control	
Description of parameters stored in contract storage (including type of parameter and whether any precision factor is applied)	
User documentation outlining on how to interact with smart contracts.	
Guideline on how to deploy the smart contract(s).	
Documentation what the (planned) initial parameters for the contract deployment/origination are/were.	

In case of inter-contract calls, which are not in scope of the security review:
• Reasoning why this contract has to be called.
• Reasoning why calling this contract is safe.
• Information about what has been done in order to be sure the called smart contract is correct and working as expected. For instance,
  • security measures in own code
  • contacting smart contract owner in order to get information about known issues
  • requesting and reviewing contract documentation
  • requesting and reviewing security review report	

Example

- https://gitlab.com/dexter2tz/dexter2tz/-/tree/liquidity_baking/docs/

Smart contract code

Item	
Code is complete	
The smart contract is fully developed and ready for review. There are no parts in the smart contract which are still under development.	
A suitable Git commit for the security assessment has been identified.	
High-level smart contract languages (e.g. SmartPy or Ligo)	
Smart contract code is appropriately documented where necessary. Comments explain tricky or core mechanics of the smart contract. Goal is that experienced contract developers / code reviewers can easily understand the code with its comments.	
Smart contract code has reasonable test coverage with meaningful test cases which are also covering possible “edge cases”. (Test cases should be “property-based”, where feasible.)	
The smart contract conforms with corresponding TZIP-standards.	
Documentation of the used compiler version (Ligo/SmartPy/etc.) to create Michelson code.	

Linters and sanity checkers, where available, are applied.	
Michelson / Tezos blockchain	
Smart contract code compiles to Michelson and can be deployed/originated on the Tezos blockchain.	
Smart contract has been deployed to a testnet and does what it's supposed to do.	
Smart contract on-chain testing, on a testnet, has been documented.	