Audit Preparing & Requirements

Audit Preparation

Audit preparation is one of the most important components of the entire security code audit process for CosmWasm. It is the process of gathering and organising the necessary information and documentation that will be used by the auditors team while conducting the security audit engagement exercise. A well-prepared audit can help to ensure that the audit is conducted efficiently and effectively, enabling the auditing team to provide an accurate and comprehensive report. This promotes further confidence in the findings and recommendations presented in the final audit report.

How to prepare for an audit

Here are some of the key steps that could assist with the audit preparation:

- **Identifying the audit scope**: The first step in audit preparation is to identify the scope of the audit and what is required to go through a comprehensive security audit. This will involve determining the specific areas of the codebase that will be audited along with any dependencies. Please provide us with access to the complete smart contract codebase, including all relevant files and dependencies. Ensure that the code is up to date and accessible in a secure repository.

- **Gathering documentation**. Any documentation related to the smart contracts, including whitepapers, technical specifications, and architectural diagrams, would be invaluable for our understanding of the project. Please share these documents with us prior to the audit. As a rule of thumb any piece of software should come with a detailed documentation to ensure its intent functionality.

➔ **Deployment Information**: If the smart contracts have already been deployed on a testnet network, please provide us with the relevant deployment details, such, contract addresses, transaction hashes and potential dapps available to interact with the components. This information will assist us in reviewing the live deployment for potential security concerns using integrating testing capabilities.

➔ **Test Cases**: If the team requesting an audit has already conducted any internal testing or security assessments on the smart contracts in-house, please share the test cases, results, and any identified issues to us. This will help us in focusing our efforts on areas that require further scrutiny.

➔ **Communicating with the auditor**: It is important to communicate with the auditor and raise any areas of extra concern throughout the audit preparation process. This will help to ensure that the auditor has all of the information they need to conduct a thorough audit. Furthermore, appoint a knowledgeable contact person to serve as the designated liaison for the audit. This individual will closely monitor the established communication channel, promptly address inquiries, and facilitate efficient and effective communication between the team and the auditors during the whole process.

Audit Preparation Checklist

- [ ] **Identify the audit scope**
  - [ ] Determine the specific areas of the codebase that will be audited.
  - [ ] Identify any dependencies that should be included in the audit.
- [ ] **Gather documentation**
  - [ ] Collect any relevant documentation related to the smart contracts, such as whitepapers, technical specifications, and architectural diagrams.
  - [ ] Ensure the documentation is comprehensive and up to date.
- [ ] **Provide access to the codebase**
  - [ ] Grant access to the complete smart contract codebase, including all relevant files and dependencies.
  - [ ] Ensure the code is accessible in a secure repository.
- [ ] **Share deployment information**
  - [ ] Provide details of the deployment, including contract addresses, transaction hashes, and any available dApps for interacting with the components.
  - [ ] If deployed on a testnet network, share the relevant deployment information.
- [ ] **Share test cases**
  - [ ] Share any test cases, results, and identified issues from internal testing or security assessments conducted in-house.
- [ ] **Communicate with the auditor**
  - [ ] Maintain open communication with the auditor throughout the preparation process.
  - [ ] Raise any areas of extra concern or provide additional information as needed.

Audit Requirements

After completing the audit preparation phase, SCV will carefully evaluate the items within the defined scope to ensure they meet the necessary requirements for proceeding with the audit process. These requirements serve as a framework to maintain consistency across audit rounds and enforce high standards of code quality. By establishing clear criteria, we aim to minimise any potential misinterpretation during the audit exercise, ensuring a comprehensive and reliable assessment within the defined scope.

To facilitate requirements, please, follow this checklist below:

Audit Requirements Checklist

- ☐ Ensure contracts are compiling and passing tests using: `cargo test`.
- ☐ Ensure `cargo fmt` is applied to better format the Rust code, ensuring consistent style and improving code quality.
- ☐ Attempt to resolve as many issues as possible from `cargo clippy`.
- ☐ Ensure there is a minimum of 40% test coverage from `cargo tarpaulin` output.
- ☐ Ensure `cargo audit` does not produce any problematic dependence when executed. This would avoid supply chain attacks.
- ☐ Ensure the code freeze hash will suffer no alteration or new commits during the audit.
- ☐ Perform code hygiene clean-up addressing the following:
    - ☐ Remove all unreachable code.
    - ☐ Remove or address all leftovers `//TODOs` if any.
    - ☐ Remove any non-relevant file templates or leftover boilerplate files.