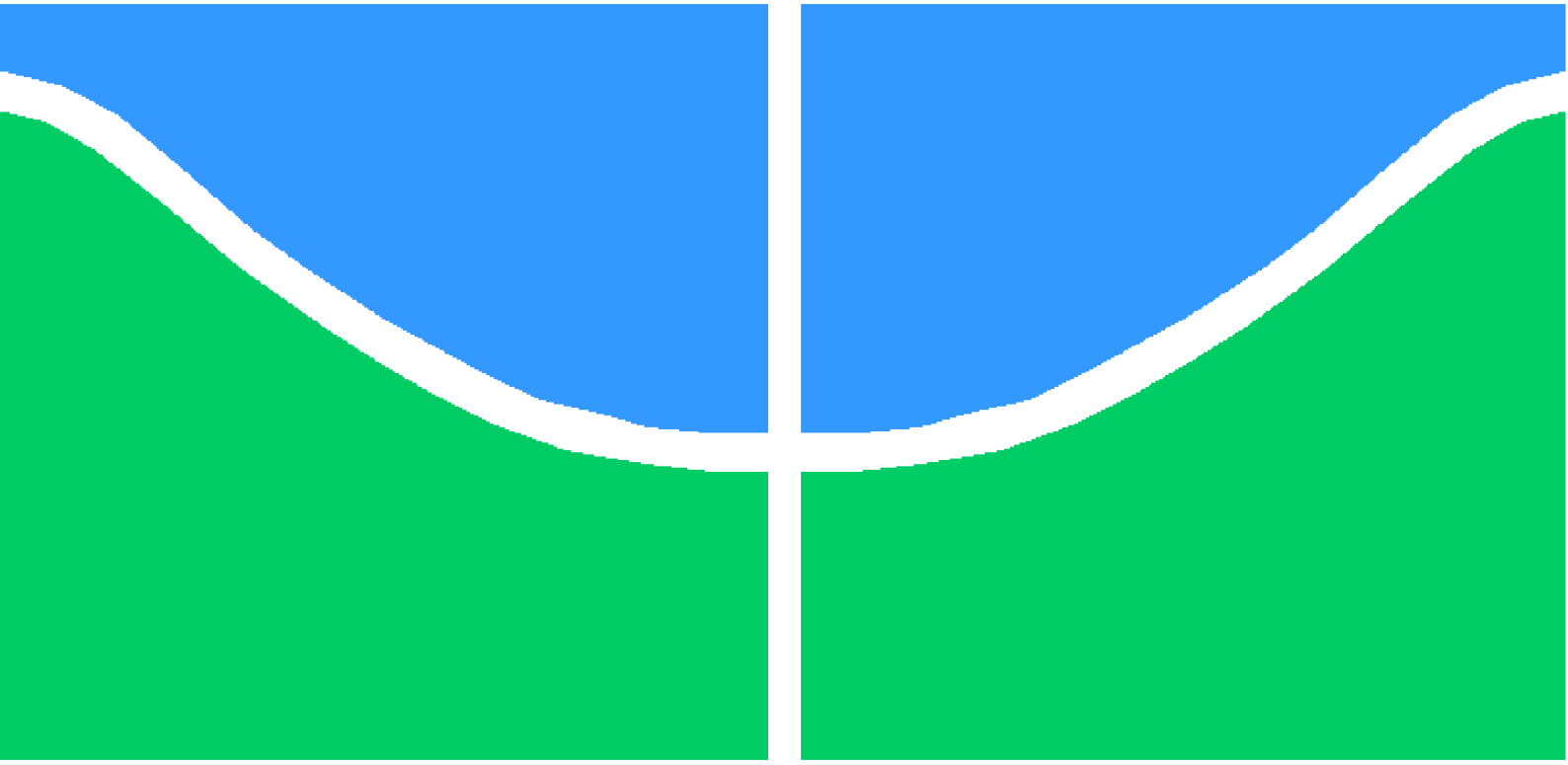

Univeristy of Brasília – UnB

Faculdade UnB Gama – FGA

Software Engineering

A Collection Of Patterns For Safe Smart Contracts

Author: Renato Britto Araujo

Counselor: Dr. Prof. Rejane Maria da Costa Figueiredo

Co-advisor: Dr. Prof. Rejane Maria da Costa Figueiredo

Brasília - DF, Brasil

2022

Renato Britto Araujo

A Collection Of Patterns For Safe Smart Contracts

Monografia submetida ao curso de graduação em Software Engineering da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Software Engineering.

Univeristy of Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Dr. Prof. Rejane Maria da Costa Figueiredo

Coorientador: Dr. Prof. Rejane Maria da Costa Figueiredo

Brasília - DF, Brasil

2022

Abstract

Key-words: Smart Contracts, Vulnerability, Patterns, Analysis

Lista de abreviaturas e siglas

UnB	University of Brasília
EVM	Ethereum Virtual Machine
NVD	A US Gov database for security vulnerabilities
CVE	Another US Gov database for security vulnerabilities

Sumário

1 Introduction

1.1 Context

Blockchain is a technology that enables trust-less communication without the need of a third party. That's achieved via a decentralized peer-to-peer network (EVM), in which its nodes (miners) are running a complete implementation of the protocols set by the blockchain as well as all the data in it. The data is in the form of a ledger, and this ledger can contain entries for raw machine code - these are known as smart contracts, compiled from the Solidity programming language, and they operate similar to how a class would: its interface exposed to the outside world via method calls. When called, the functions are executed for a fee paid to the nodes that execute the computation separately. Smart contracts enable enforcement of agreed terms between two or more untrusted parties and once one is deployed, it cannot be revoked and becomes a permanent part of the ledger. Finally, a smart contract may hold real money just like a class may hold an integer or string. Because of all these properties, smart contracts' code errors and vulnerabilities can be disastrous, and prevention against them a must.

1.2 Literature Review

Several papers tackle security risks and protection for smart contracts. In (CHEN et al., 2020), (KUSHWAHA et al., 2022) and (SAYEED; MARCO-GISBERT; CAIRA, 2020), an extensive study and analysis is performed over existing literature and vulnerability databases (such as NVD or CVE) to detect and categorize them with rigor. Furthermore, (SINGH et al., 2020), (ALI; ABIDEEN; ULLAH, 2021) and (VIVAR; OROZCO; VILLALBA, 2021) addresses approaches to automated and manual detection and frameworks for designing secure smart contracts. asdasd

1.3 Problem

Smart contract programming is a new field, with little to no patterns, which enable the developers to creatively (and dangerously) design their contracts. Some designs for a smart contract are safer than others, and contracts made with ease of testing and understanding are more likely to successfully avoid corruption or having funds robbed. Despite several audits, the more complex a project is, the more likely a vulnerability is to be found, so safety is a matter of caution for the smallest to biggest institutions

creating smart contracts. Several automated vulnerability detection tools exist, but the architecture of a contract can fail regardless.

1.4 Objective

With the intent to provide smart contract programmers a set of canned, battle-tested smart contract design patterns, this study identifies, compiles and investigates multiple contracts using several analysis frameworks and well-known vulnerability lists to extract similarities in smart contracts which tend to yield higher security independent of their domain.

The work is composed of several parts which construct on top of the other:

1. **Vulnerability Analysis Accumulation:** firstly, this research accumulate a body of vulnerabilities and frameworks for detecting them. To construct this, a systematic literature review of vulnerabilities and vulnerability prevention methods is performed targeting the question: *what makes a smart contract unsafe?*
2. **Sample Gathering**: through online platforms in which open-source repositories can be found, such as Github or Gitlab, a list of well-known and frequently used smart contracts, through which a significant amount financial assets are managed, is compiled. The reason for such constraints is to ensure that by their exposure to a wide audience and real rewards for exploitation, their continued untroubled operation means they are safer and the large work that was put into them gets analysed for further usage.
3. **Pattern Identification**: via cross-referencing analysis over the smart contracts, shared patterns should naturally emerge. These patterns will be found and listed in relation to the problem it tries to solve on a abstract level.
4. **Pattern Testing**: A final list of patterns is tested against the analysis framework on step 1, to evaluate their security when solving a specific common need. These patterns also get evaluated on other practical dimensions such as scalability and price.

A non-goal for this paper, therefore, is not to accumulate the latest research and discoveries related to smart contract vulnerabilities and prevention methods, but to identify patterns in smart-contracts that have not yet been successfully exploited after it's had ample chance of such an event happening.

1.5 Paper Organization

This paper is organized as follows:

- **Section 1 - Introduction:** The context, research problem and objective, and a methodological synthesis;
- **Section 2 - Related Work:** Includes works that contribute to the conclusions of this paper.
- **Section 3 - Methodology:** Studies the methodological route taken in depth.
- **Section 4 - Investigation:** The information gathering process, pattern identification and analytical testing.
- **Section 4 - Proposals:** Presents patterns for safe smart contract development.

2 Related Work

2.1 Initial Considerations

In this chapter, an analysis of the body of knowledge

2.2 Vulnerabilities

2.3 Vulnerability Prevention

2.4 Final Considerations

3 Materiais e Métodos

3.1 Considerações Iniciais

Neste capítulo apresenta-se o plano metodológico adotado para alcançar o objetivo desta pesquisa, isto é...

3.2 Plano Metodológico

O plano metodológico adotado possui 4 fases: Planejamento de pesquisa; Coleta de dados; Análise de dados; e Resultados...

3.2.1 Planejamento

Nesta fase foram definidos o tema de pesquisa, a questão de pesquisa, o objetivo a ser atingido e a classificação metodológica.

3.2.2 Coleta de Dados

Nesta fase foram adotadas técnicas como *Pesquisa Bibliográfica* e *Design Science Research* para a coleta de dados...

3.2.3 Analise de Dados

Nesta fase é feita uma analise nos resultados observados nas fases anteriores.

3.2.4 Resultados

Esta fase corresponde à fase final deste trabalho, na qual os resultados obtidos com a execução do Trabalho de Conclusão 2 serão apresentados.

3.3 Considerações Finais

Neste capítulo, foi apresentado o plano metodológico adotado para se atingir os objetivos desta pesquisa. No próximo capítulo apresenta-se a proposta deste trabalho, com as atividades já realizadas e as que ainda serão realizadas.

4 Proposta do Trabalho

4.1 Considerações Iniciais

Neste capítulo é retomado o plano metodológico apresentado brevemente no Capítulo 3 e apresenta um detalhamento da proposta deste trabalho bem como as atividades já realizadas e as atividades a serem realizadas...

4.2 Planejamento de Pesquisa

Nesta fase foram definidos o tema de pesquisa, a questão de pesquisa, o objetivo a ser atingido e a classificação metodológica. O Capítulo 1 contempla esta fase.

4.3 Coleta de Dados

Nesta fase foram adotadas técnicas como Pesquisa Bibliográfica e *Design Science Research* para a coleta de dados.

4.3.1 Pesquisa Bibliográfica

4.4 Cronograma

5 Results

6 Analysis

7 Conclusion and Future Work

Referências

ALI, A.; ABIDEEN, Z. U.; ULLAH, K. Sescon: secure ethereum smart contracts by vulnerable patterns' detection. *Security and Communication Networks*, Hindawi, v. 2021, 2021. Citado na página 5.

CHEN, H. et al. A survey on ethereum systems security: Vulnerabilities, attacks, and defenses. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 53, n. 3, p. 1–43, 2020. Citado na página 5.

KUSHWAHA, S. S. et al. Systematic review of security vulnerabilities in ethereum blockchain smart contract. *IEEE Access*, IEEE, 2022. Citado na página 5.

SAYEED, S.; MARCO-GISBERT, H.; CAIRA, T. Smart contract: Attacks and protections. *IEEE Access*, IEEE, v. 8, p. 24416–24427, 2020. Citado na página 5.

SINGH, A. et al. Blockchain smart contracts formalization: Approaches and challenges to address vulnerabilities. *Computers & Security*, Elsevier, v. 88, p. 101654, 2020. Citado na página 5.

VIVAR, A. L.; OROZCO, A. L. S.; VILLALBA, L. J. G. A security framework for ethereum smart contracts. *Computer Communications*, Elsevier, v. 172, p. 119–129, 2021. Citado na página 5.