Análise de Bugs em Contratos Inteligentes de Blockchains Compatíveis com Ethereum Virtual Machine: Janeiro a Setembro de 2023

Ana Julia Bittencourt Fogaça

[1]Universidade do Sul de Santa Catarina (UNISUL)
Tubarão - SC - Brasil
anajuliabit@gmail.com

1. Abstract

Software is developed by humans and, therefore, is inherently subject to flaws and bugs. In the domain of decentralized applications (dApps) operating on the Ethereum Virtual Machine (EVM), these vulnerabilities take on a critical dimension. Unlike conventional software, dApps represent attractive targets for hackers, due to the transparent and open-source nature of blockchains. This article provides a comprehensive analysis of 145 bugs identified in public audits of various projects developed in Solidity and hosted on Ethereum or on EVM-compatible blockchains. Through this analysis, we seek to provide valuable insights into the nature and frequency of vulnerabilities in smart contracts, highlighting the importance of robust development and auditing practices in the blockchain ecosystem.

2. Resumo

Os softwares são desenvolvidos por humanos e, por isso, estão inerentemente sujeitos a falhas e bugs. No domínio das aplicações descentralizadas (dApps) que funcionam na Ethereum Virtual Machine (EVM), essas vulnerabilidades adquirem uma dimensão crítica. Diferentemente dos softwares convencionais, as dApps representam alvos atraentes para hackers, devido à natureza transparente e de código aberto das blockchains. Este artigo oferece uma análise abrangente de 145 bugs identificados em auditorias públicas de vários projetos desenvolvidos em Solidity e hospedados na Ethereum ou em blockchains compatíveis com a EVM. Através desta análise, buscamos fornecer insights valiosos sobre a natureza e a frequência de vulnerabilidades em contratos inteligentes, destacando a importância de práticas de desenvolvimento e auditoria robustas no ecossistema blockchain.

3. Introdução

Introduzida em 2008 pelo whitepaper do Bitcoin [15], a tecnologia blockchain é reconhecida como um vetor de transformação em diversas indústrias [22]. Caracterizada por sua segurança robusta, transparência e rastreabilidade, além da natureza de código aberto, a blockchain tem encontrado aplicação em operações críticas de negócios. Até 2023, seu uso mais notável, o mercado de criptomoedas, alcançou um valor de mercado superior a um trilhão de dólares [6]. A aplicabilidade da blockchain estende-se além das criptomoedas, impactando setores como finanças, gerenciamento de cadeia de suprimentos, identidade, saúde e governança eleitoral [2].

A inovação trazida pelo Ethereum, lançado com seu whitepaper em 2014, marcou um ponto de virada na evolução da blockchain. Diferentemente do Bitcoin, que foi concebido primariamente como uma moeda eletrônica peer-to-peer [15], o Ethereum introduziu o conceito revolucionário de "contratos inteligentes"[8]. Essa funcionalidade expandiu o escopo de aplicação da blockchain para novas áreas. A plataforma Ethereum destaca-se por sua máquina virtual, capaz de executar códigos em linguagens Turing-complete[24]. No entanto, como os contratos inteligentes são criados por humanos, eles não estão isentos de falhas. Em um ambiente de código aberto, típico de blockchains como Ethereum, essas falhas podem se tornar alvos atraentes para hackers. Somente no primeiro trimestre de 2023, ataques à rede Ethereum resultaram no roubo de 320 milhões de dólares [11]. Para mitigar tais riscos, auditorias de contratos inteligentes, sejam privadas, conduzidas por empresas especializadas, ou públicas, através de plataformas como Code4rena [4] e Sherlock [18], são práticas comuns, onde vulnerabilidades podem ser identificadas e os descobridores recompensados.

Desde 2020, projetos de blockchain que negligenciaram o processo de auditoria sofreram perdas financeiras significativas, estimadas em 3.69 bilhões de dólares, enquanto projetos auditados reportaram perdas de 1.3 bilhões [5]. Isso indica que, embora as auditorias reduzam a probabilidade de ataques bem-sucedidos, ainda existem desafios na detecção precoce de vulnerabilidades. Com a demanda crescente por contratos inteligentes e uma projeção de aumento anual de 82,2% de 2023 a 2030 [19], torna-se crucial compreender e classificar as vulnerabilidades emergentes. Neste estudo, analisamos 145 bugs identificados em 31 competições de auditoria públicas realizadas entre janeiro e setembro de 2023 nas plataformas renomadas Sherlock [18] e Code4rena [4]. Nosso objetivo é elucidar aspectos fundamentais, como a complexidade na detecção de diferentes tipos de bugs e a incidência de categorias específicas de bugs em variados tipos de protocolos.

O artigo está estruturado da seguinte forma: inicialmente, apresentamos uma revisão bibliográfica, abordando os conceitos de blockchain, contratos inteligentes, EVM, Solidity e Finanças descentralizadas. Em seguida, descrevemos a metodologia empregada, a taxonomia e as categorias dos protocolos analisados. Prosseguimos com uma análise dos dados coletados e, por fim, discutimos os resultados obtidos.

4. Revisão bibliográfica

4.1. Ethereum Blockchain

A tecnologia Blockchain, particularmente a Blockchain Ethereum, representa um avanço significativo no âmbito dos sistemas descentralizados. Uma blockchain é essencialmente um banco de dados distribuído que mantém uma lista continuamente crescente de registros, chamados blocos, que são vinculados e protegidos usando criptografia. Cada bloco geralmente contém um hash criptográfico do bloco anterior, um carimbo de data/hora e dados de transação, tornando-o resistente à modificação de dados [12].

Ethereum, lançado em 2015, vai além das funcionalidades básicas de um blockchain. É uma plataforma descentralizada que permite a criação e execução de aplicativos decentralizados sem tempo de inatividade, fraude, controle ou interferência de terceiros [16]. A blockchain do Ethereum não registra apenas transações, mas também o estado de cada contrato inteligente, tornando-o um blockchain mais versátil e programável em comparação com seu antecessor, o Bitcoin [10].

4.2. Contratos inteligentes

Contratos inteligentes representam programas autônomos executados em uma blockchain. Fundamentalmente, um contrato inteligente é um aplicativo que modifica o estado da blockchain quando determinadas condições pré-definidas são satisfeitas, eliminando a necessidade de intermediários. Uma vez implementados na blockchain, os contratos inteligentes são imutáveis, o que implica a necessidade de uma análise rigorosa e minuciosa do código para identificar e corrigir potenciais vulnerabilidades antes de sua efetivação. Além disso, contratos inteligentes têm a capacidade de interagir uns com os outros, facilitando o desenvolvimento de aplicações complexas, como as que são encontradas nas Finanças Descentralizadas (DeFi) [25].

4.3. Ethereum Virtual Machine

A Ethereum Virtual Machine (EVM) é o núcleo da plataforma Ethereum, funcionando como um ambiente de execução global e descentralizado para contratos inteligentes. A EVM é uma máquina de estado que executa contratos inteligentes na rede Ethereum, permitindo que desenvolvedores criem aplicações que se beneficiam da criptografia e da descentralização. Cada nó na rede Ethereum executa uma instância da EVM, garantindo que o estado da blockchain Ethereum seja mantido uniforme e sincronizado em toda a rede. A EVM é notável por sua Turing-completude, o que significa que, teoricamente, ela pode executar qualquer algoritmo, desde que haja recursos computacionais suficientes [14].

4.4. Solidity

Solidity é uma linguagem de programação de alto nível para a implementação de contratos inteligentes e é fortemente tipada, suporta herança, bibliotecas e tipos de usuário complexos[20]. Projetada para se alinhar com a EVM, Solidity facilita o desenvolvimento de contratos inteligentes através de uma sintaxe semelhante a JavaScript, tornando-a acessível a um amplo espectro de programadores. Solidity, apesar de ser uma linguagem de alto nível com características robustas, não está isenta de vulnerabilidades. Muitas delas decorrem de uma desconexão entre a semântica da linguagem e a intuição dos programadores, principalmente porque Solidity implementa características de linguagens conhecidas, como JavaScript, de maneiras peculiares. Além disso, a linguagem carece de construções específicas para lidar com aspectos do domínio de blockchain, como a imprevisibilidade na ordem ou no atraso das etapas de computação registradas publicamente na blockchain. Isso ressalta a importância de uma compreensão aprofundada de Solidity ao desenvolver contratos inteligentes, para mitigar o risco de vulnerabilidades de segurança.

4.5. Finanças Descentralizadas

Finanças Descentralizadas (DeFi) constituem uma infraestrutura financeira inovadora, fundamentada na tecnologia blockchain. Utilizando contratos inteligentes, DeFi busca replicar e aprimorar serviços financeiros tradicionais, oferecendo maior abertura, interoperabilidade e transparência [17]. Este segmento emergente, que se alinha com as tendências de FinTech, RegTech, criptomoedas e ativos digitais, está rapidamente ganhando relevância. No entanto, seu significado integral, implicações legais e impactos políticos ainda estão em fase de compreensão e análise [9].

DeFi representa um movimento revolucionário na direção de um sistema financeiro operado inteiramente por código, eliminando intermediários tradicionais. Entre 2020 e 2023, o setor experimentou um crescimento exponencial, de menos de 1 bilhão de dólares para um impressionante patamar de 45 bilhões [7]. Esse crescimento sublinha a complexidade e a necessidade de uma compreensão aprofundada de suas nuances.

Como um novo paradigma na tecnologia financeira, DeFi tem o potencial de reestruturar a arquitetura da finança moderna, abrindo novas avenidas para empreendedorismo e inovação. Este setor promissor enfrenta desafios e limitações significativos, mas oferece oportunidades sem precedentes para remodelar o futuro das finanças [3].

5. Metodologia

5.1. Categoria dos protocolos

Os protocolos investigados neste estudo são dedicados ao setor de DeFi, exclusivamente as seguintes subcategorias conforme a classificação proposta por DefiLlama [7]:

- Derivativos: Protocolos que disponibilizam ferramentas para operações financeiras alavancadas, possibilitando que os usuários façam previsões e especulações acerca de valores futuros de ativos, amplificando suas projeções de lucro ou prejuízo.
- Yield Farming: Protocolos que incentivam a prática de staking ou fornecimento de liquidez por parte dos usuários, oferecendo recompensas por tais atividades.
- Agregadores de Yield: Protocolos que otimizam os rendimentos por meio da integração de diversas estratégias de *yield farming*.
- Opções: Protocolos que ofertam o direito, embora não a obrigação, de adquirir um ativo por um valor preestabelecido em um momento futuro.
- DAOs (Organizações Autônomas Descentralizadas): Entidades organizacionais inovadoras que operam sem centralização, com decisões sendo tomadas de forma coletiva pelos membros.
- Launchpads: Protocolos desenvolvidos para lançar novos projetos e criptoativos no mercado.
- Índices: Protocolos que rastreiam ou replicam a performance de uma série de ativos interligados.
- DEXs (Trocas Descentralizadas): Protocolos que permitem a troca de criptoativos de forma descentralizada.
- RWAs (Ativos do Mundo Real): Protocolos relacionados à tokenização de ativos físicos, como imóveis.
- Stablecoins: Criptomoedas com valor atrelado a moedas fiduciárias ou outros ativos, buscando manter sua estabilidade por intermédio de mecanismos descentralizados.
- Gestores de Liquidez: Protocolos que gerenciam posições de liquidez em formadores de mercado automatizados com liquidez concentrada.
- Empréstimos: Protocolos que permitem o empréstimo e a tomada de empréstimos de diversos ativos.

5.2. Classificação dos Bugs

A classificação das vulnerabilidades dos protocolos analisados neste trabalho segue uma taxonomia híbrida, combinando os modelos propostos por Zhang et al. [26] e as tags de Solodit [21], detalhada da seguinte forma:

- O: Fora do Escopo
  - Inacessibilidade ao código-fonte do projeto.
  - Bugs manifestados em componentes off-chain.
  - Contratos inteligentes desenvolvidos em linguagens distintas
- C01: Manipulação do Mempool / Vulnerabilidades de Front-Running
  - Ataques do tipo sandwich
  - Exploração baseado em *flash loans*
- C02: Ataque de Reentrada. Vulnerabilidades de reentrância, resultantes de chamadas externas realizadas antes da conclusão de atualizações de estado internas, possibilitando a um adversário explorar o estado inconsistente.
- C03: Atualizações de Estado Errôneas. Ausência ou incorreção na atualização de estado, tal como uma atualização que não deveria ser efetuada.
- C04: Configuração *Hardcoded*. Inserção de valores ou parâmetros estáticos diretamente no código do contrato inteligente, o que pode representar um risco de segurança se houver necessidade de flexibilidade.
- C05: Escalada de Privilégios e Problemas de Controle de Acesso.
  - Chamada de funções privilegiadas sem restrições adequadas.
  - Fundos de usuários que podem ser imobilizados por falhas ou ausência de código de retirada.
- C06: Matemática Incorreta / Contabilidade Errônea. Erros de cálculo decorrentes de implementações matemáticas falhas, conduzindo a resultados imprecisos, incluindo:
  - Ordem incorreta de operações.
  - Retorno de valores inesperados.
  - Utilização de números incorretos para cálculos.
  - Erros de contabilidade.
  - Underflow/overflow.
- C07: Lógica de Negócios Quebrada. Defeitos na lógica de negócios ou protocolos que, mesmo alinhados à intenção do desenvolvedor, são inerentemente falhos.
  - Invocações de funções inesperadas ou omissas
  - Condições anômalas de ambiente ou contrato
  - Argumentos de função impróprios
- C08: Bugs Específicos da Implementação do Contrato. Bugs que não se enquadram claramente em outras categorias.
- C09: Falta de Proteção Contra Replay de Assinatura
  - Nonce ausente
  - Colisão de hash
- C10: Verificação Ausente. Omissão crítica de condições ou validações essenciais no código.
- C11: DoS (Negação de Serviço). Vulnerabilidades que permitem a um atacante comprometer a resposta ou eficiência do contrato. Esta categoria inclui casos que não são bem descritos por outra classe e onde a consequência primária é o encerramento do contrato ou ineficiência operacional.
- C12: Validação de Dados. Falhas na verificação ou saneamento de entradas, particularmente daquelas oriundas de fontes externas.
- C13: Correspondência de Lista Branca/Lista Negra. Gerenciamento inadequado de endereços baseado em listas predefinidas.

- C14: Arrays. Vulnerabilidades associadas ao manuseio inadequado de arrays, incluindo:
  - Leituras/escritas fora dos limites
  - Problemas na exclusão
  - Questões relacionadas ao redimensionamento de arrays

5.3. Coleta de dados

Tabela 1. Visão Geral das Competições Avaliadas. 'HRF' (High Risk Findings) representa bugs de alta severidade. 'nSLOC' indica o número total de linhas de código associadas a cada competição.

Plataforma	Categoria	Competição	Prêmio	HRF	nSLOC	Par
Code4rena	DAO	Arbitrum security council election	90500	1	2184	
Code4rena	DAO	Llama	60500	2	2096	
Code4rena	Stablecoin	Lybra finance	60500	8	1762	
Code4rena	Dexes	Maia DAO ecosystem	300500	35	10997	
Code4rena	Yield	PoolTogether	121650	9	3324	
Code4rena	Yield	PoolTogether v5: part deux	42000	2	1001	
Sherlock	Lending	Ajna update	85600	6	5659	
Sherlock	Yield Agreggator	Blueberry	72500	10		
Sherlock	Yield Agreggator	Blueberry Update #3	23600	5	3633	
Sherlock	Opções	Bond options	23600	2	885	
Sherlock	Empréstimos	Cooler update	17000	4	512	
Sherlock	Dexes	GFX labs	20400	2	716	
Sherlock	Derivativos	GMX	200000	5	10571	
Sherlock	Lending	Iron bank	67400	1	2241	
Sherlock	Derivativos	Perennial	122000	1	4063	
Sherlock	Derivativos	Perennial v2	125200	6	2494	
Sherlock	Derivativos	Symmetrical	91000	8	3553	
Sherlock	Derivativos	Symmetrical Update	27600	2	3921	
Sherlock	Launchpad	Tokensoft	21400	1	769	
Sherlock	Stablecoin	Unitas protocol	36400	1	1433	
Code4rena	RWA	Ondo finance	60500	1	4365	
Sherlock	Índices	Index coop	130600	2	4383	
Sherlock	Stablecoin	USSD	18200	3	402	
Sherlock	RWA	Dinari	16000	1	575	
Sherlock	Dexes	RealWagmi	33200	5	1080	
Code4rena	DAO	Nouns DAO	100000	1	9098	
Sherlock	Dexes	DODO v3	57800	5	2079	
Sherlock	Derivativos	Hubble Exchange	60000	3	1945	
Code4rena	Stablecoin	Angle Protocol	52500	3	2276	
Code4rena	Gestores de liquidez	Arrakis	81400	2	2801	
Sherlock	Dexes	Unstoppable	36400	8	2035	
TOTALS			2255950	145	3095.1	1

No período de janeiro a setembro de 2023, nossa análise focou em 31 das competições de auditoria pública realizadas nas plataformas Code4rena [4] e Sherlock [23], nas

quais foram identificados 145 bugs de alta severidade. Com duração média de sete dias, estes eventos visaram a identificação de falhas críticas em contratos inteligentes antes de seu lançamento final. A Tabela 1 detalha as competições examinadas, incluindo o número de participantes, o prêmio total, a categoria do projeto, a quantidade de vulnerabilidades de alta severidade encontradas e o total de linhas de código abrangidas em cada competição. O valor total das recompensas distribuídas nesses eventos superou os dois milhões de dólares, com uma média de 150 participantes por evento. Após a fase de submissão, juízes especializados em auditoria de contratos inteligentes, selecionados pela comunidade, avaliaram a severidade dos bugs. Os bugs classificados como de alta severidade representam riscos significativos, incluindo a possibilidade de roubo ou perda de ativos digitais [13]. Todos os bugs, suas descrições e a classificação realizada estão disponíveis no nosso dataset [1]

6. Análise e Discussão dos Resultados

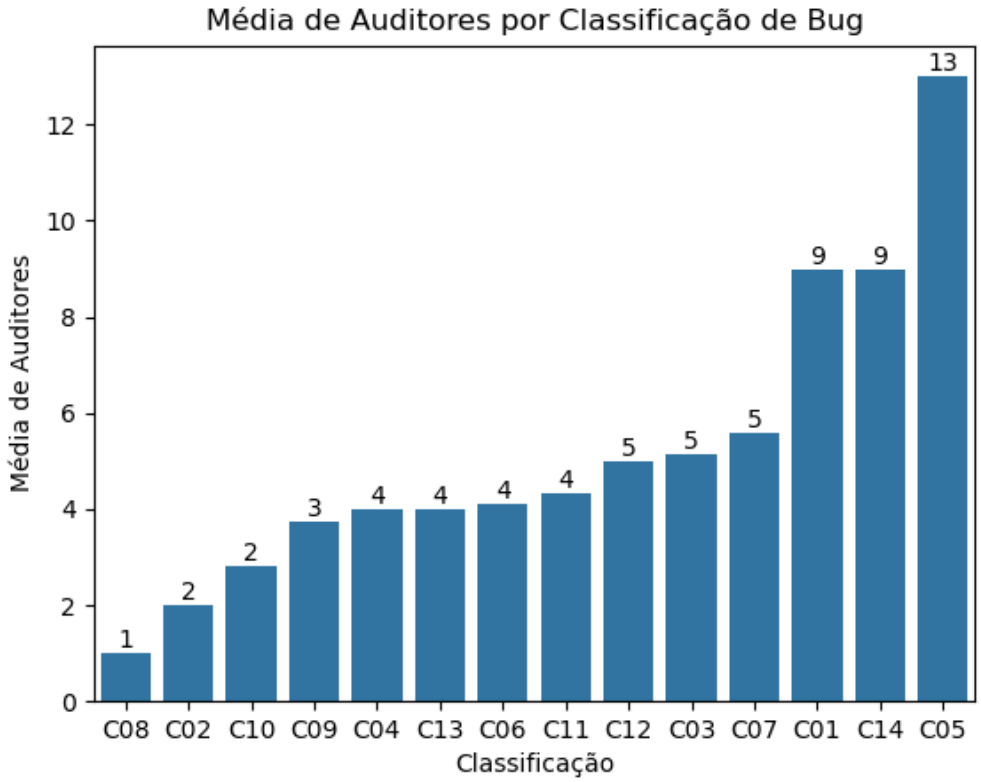

Figura 1. Média de auditores por classificação

6.1. Análise dos dados coletados

A identificação de vulnerabilidades em contratos inteligentes é uma tarefa que exige extrema atenção aos detalhes e uma mentalidade orientada para a descoberta de falhas, similar à de um potencial atacante. Neste contexto, auditores enfrentam desafios significativos na condução de suas investigações, especialmente porque muitos bugs permanecem não detectados [5]. Para aprofundar nosso entendimento desses desafios, conduzimos uma

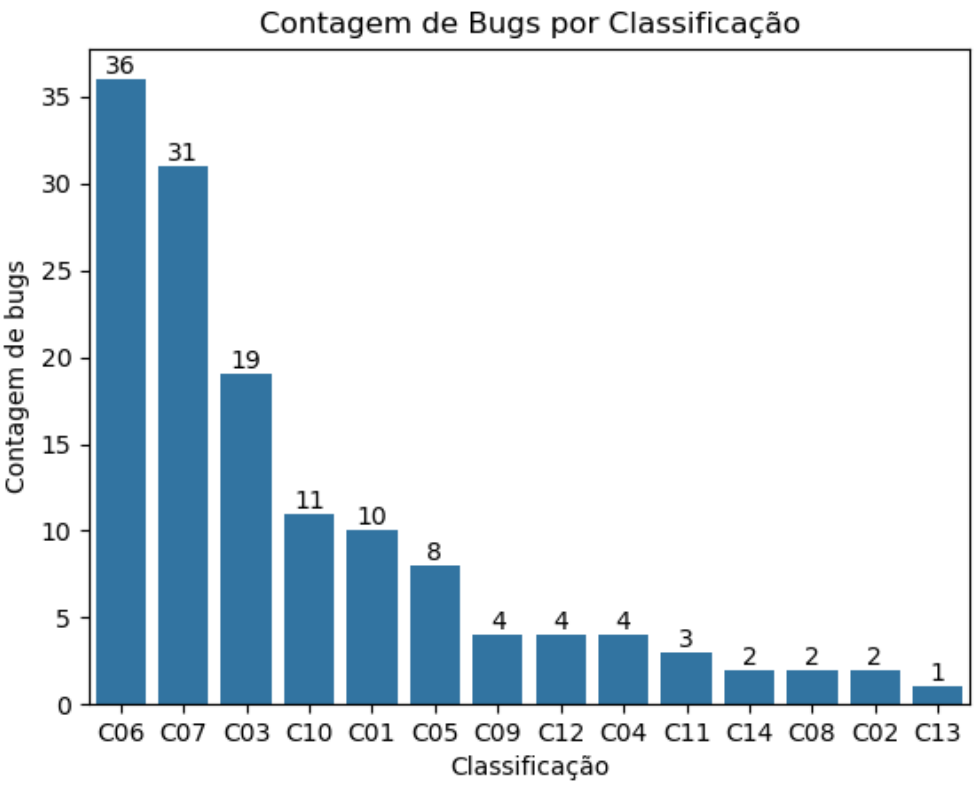

Figura 2. Contagem de Bugs por Classificação

análise exploratória do dataset, utilizando as bibliotecas do Python, Panda e Seaborn. O script utilizado nessa análise está documentado em [1], e todos os gráficos nesta seção foram gerados a partir dele.

Nossas principais descobertas incluem:

1. A detecção de bugs por um número reduzido de auditores sugere uma complexidade elevada em sua identificação. Como ilustrado na Figura 1, entre as diversas categorias de vulnerabilidades, aquelas relacionadas a Falhas Específicas na Implementação de Contratos (C08), seguidas por Vulnerabilidades de Reentrância (C02) e Ausência de Verificações (C10), emergem como as mais desafiadoras. Em contraste, categorias como Problemas de Controle de Acesso e Escalada de Privilégios (C05), Arrays (C14), e Manipulação de Mempool / Vulnerabilidades de Front-Running (C01) são mais facilmente identificáveis.
2. C06 - Erros de Cálculo (Matemática Incorreta/Contabilidade Errônea): Conforme a Figura 2, essa categoria é a mais prevalente, indicando que falhas em lógica matemática, como underflows/overflows ou sequências erradas de operações, são comuns no setor de Finanças Descentralizadas (DeFi). Isso é compreensível, considerando a complexidade dos cálculos matemáticos no DeFi. Essa observação sugere que muitos contratos inteligentes são vulneráveis devido a implementações inadequadas de operações matemáticas, resultando potencialmente em perdas financeiras significativas.
3. C07 - Lógica de Negócios Quebrada: Esta é a segunda classificação mais comum,

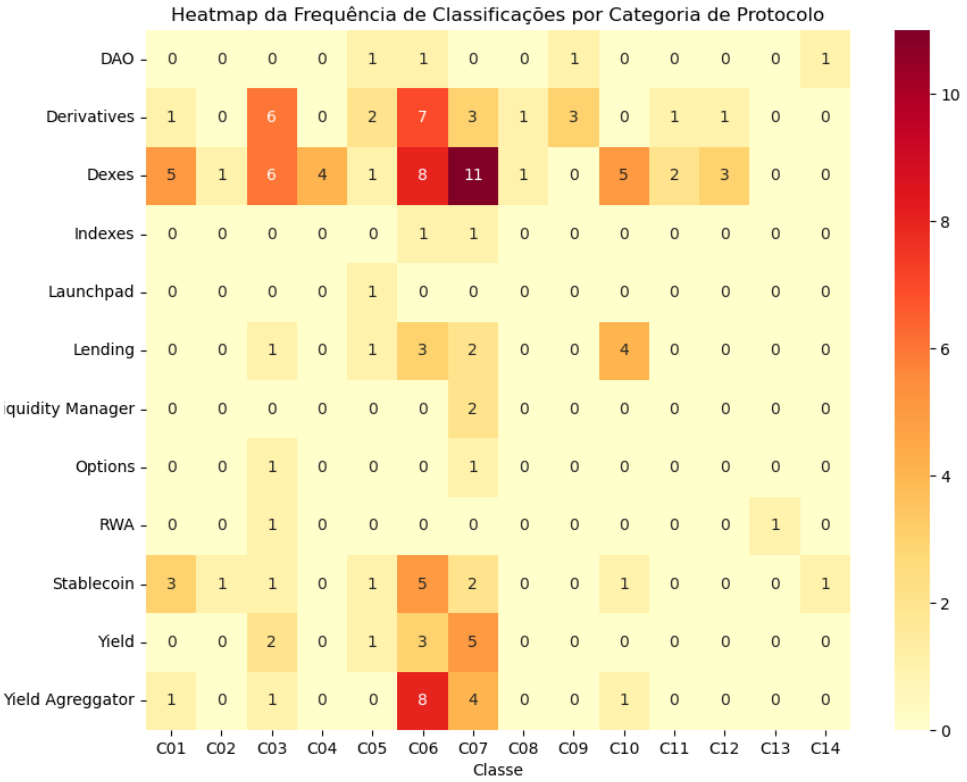

Figura 3. Frequência de Classificações por Categoria de Protocolo

como mostra a Figura 2, indicando que muitos contratos têm falhas na lógica de negócios, que poderiam ser mitigadas com análises e testes mais detalhados das regras de negócio e cenários de contrato.

4. C03 - Atualizações de Estado Errôneas: Esta classificação, a terceira em frequência, sugere que uma parcela significativa de contratos possui falhas na lógica que gerencia o estado do contrato, levando a consequências imprevistas e possíveis vulnerabilidades de segurança.
5. Classes Menos Frequentes (C02, C08, C13): Vulnerabilidades como Reentrância (C02) e Falhas Específicas na Implementação de Contratos (C08) são menos frequentes, conforme a Figura 2, o que pode indicar maior dificuldade de detecção, alinhando-se com a descoberta 1 de que C02 e C08 são categorias desafiadoras. A classe C13, que inclui vulnerabilidades como o uso de funções de hash inseguras, também é menos frequente.
6. Classes dominantes de vulnerabilidades, como Lógica de Negócios Quebrada (C07), Atualizações de Estado Incorretas (C03), Problemas de Controle de Acesso e Escalada de Privilégios (C05), e Erros de Cálculo (C06), são recorrentes em várias categorias, conforme indica a Figura 3. Esses padrões reforçam a prevalência destes tipos de vulnerabilidades.
7. Diferentes categorias de protocolo apresentam distintos perfis de vulnerabilidade. Protocolos Dexes registraram o maior número de bugs, com 34.3% dos bugs

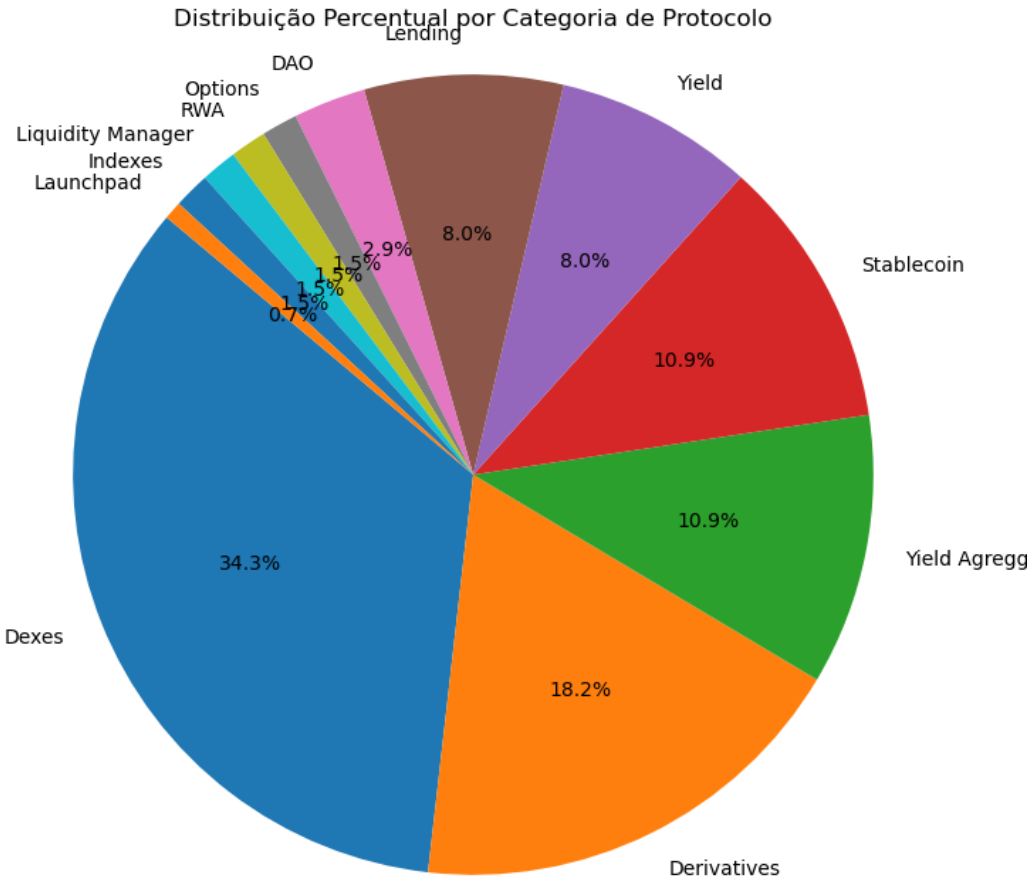

Figura 4. Distribuição Percentual por Categoria de Protocolo

identificados pertencendo a esta categoria, seguido por Derivativos com 18.2%, e Agregadores e Stablecoins com aproximadamente 10% cada, conforme ilustrado na Figura 4.

6.2. Discussão dos Resultados

Esta seção visa aprofundar a análise dos resultados obtidos e explorar suas implicações práticas

1. Complexidade na Identificação de Vulnerabilidades: Os resultados revelam uma complexidade considerável na detecção de certas categorias de vulnerabilidades, especialmente aquelas relacionadas a falhas específicas na implementação de contratos e vulnerabilidades de reentrância. Isso enfatiza a necessidade de ferramentas e métodos de auditoria mais sofisticados, assim como a importância de uma formação contínua para desenvolvedores e auditores, a fim de equipá-los com as habilidades necessárias para identificar tais vulnerabilidades.

2. Foco na Verificação de Lógica Matemática: A prevalência de erros de cálculo no DeFi sugere a necessidade de uma maior atenção na verificação de lógicas matemáticas e operações contábeis. Práticas como testes de fuzzing, revisões de código por pares e simulações podem ser essenciais para prevenir falhas.
3. Vulnerabilidades na Lógica de Negócios e Atualizações de Estado: A frequência dessas categorias de vulnerabilidades aponta para a necessidade de uma análise mais profunda da lógica de negócios e dos mecanismos de atualização de estado nos contratos inteligentes. Este aspecto sublinha a importância de um design de contrato cuidadoso, além de testes abrangentes e análises de cenários para garantir a integridade do contrato.
4. Diferentes Perfis de Vulnerabilidade em Diferentes Protocolos: A análise destacou variações significativas nos perfis de vulnerabilidade entre diferentes tipos de protocolos de blockchain, como Dexes, Derivativos e Stablecoins. Isso implica que estratégias de segurança e auditoria devem ser adaptadas conforme o tipo de protocolo, considerando suas características e riscos específicos.

7. Considerações finais

Este estudo analisa 145 bugs identificados em competições de auditorias públicas, destacando aspectos cruciais e desafios relacionados à segurança de contratos inteligentes. As descobertas principais indicam a complexidade na identificação de vulnerabilidades específicas, sublinhando a necessidade de ferramentas e habilidades de auditoria avançadas. Notavelmente, a prevalência de erros de cálculo no setor de DeFi e falhas na lógica de negócios ressalta a importância de verificar rigorosamente as operações matemáticas e analisar detalhadamente a lógica de negócios. Os resultados enfatizam a necessidade de uma abordagem holística e meticulosa no desenvolvimento e na auditoria de contratos inteligentes, destacando a importância de ferramentas de análise mais sofisticadas, educação contínua para desenvolvedores e auditores, e a adoção de práticas rigorosas de teste e validação. Para futuras pesquisas, recomenda-se a exploração de métodos de inteligência artificial e aprendizado de máquina na detecção e análise de vulnerabilidades. Estas tecnologias podem oferecer uma análise mais aprofundada e abrangente, identificando padrões complexos e sutilezas. Além disso, a colaboração entre comunidades acadêmicas, desenvolvedores de blockchain e profissionais de segurança cibernética é sugerida para estabelecer benchmarks e frameworks padronizados para a avaliação de segurança em contratos inteligentes. Essa colaboração pode levar ao desenvolvimento de melhores práticas, diretrizes de codificação segura e ferramentas de auditoria mais robustas, contribuindo significativamente para a segurança no ecossistema de Finanças Descentralizadas (DeFi).

Referências

[1] Ana Julia Bittencourt. *Solidity Common Vulnerabilities*. Out. de 2023. URL: https://github.com/anajuliabit/solidity-common-vulnerabilities (acesso em 12/11/2023).

[2] *Blockchain Adoptions in the Maritime Industry: A Conceptual Framework*. URL: https://www.tandfonline.com/doi/epdf/10.1080/03088839.2020.1825855?needAccess=true (acesso em 06/10/2023).

[3] Yan Chen e Cristiano Bellavitis. “Blockchain Disruption and Decentralized Finance: The Rise of Decentralized Business Models”. Em: *Journal of Business Venturing Insights* 13 (jun. de 2020), e00151. ISSN: 2352-6734. DOI: 10.1016/j.jbvi.2019.e00151. URL: https://www.sciencedirect.com/science/article/pii/S2352673419300824 (acesso em 09/11/2023).

[4] *Code4rena | Keeping High Severity Bugs out of Production*. URL: https://code4rena.com/ (acesso em 01/11/2023).

[5] *Competitive vs Private Audits - Pros and Cons | The Full Comparison*. URL: https://www.cyfrin.io/blog/competitive-vs-private-audits-comparison (acesso em 09/11/2023).

[6] *Cryptocurrency Statistics 2023*. Set. de 2023. URL: https://www.forbes.com/advisor/au/investing/cryptocurrency/cryptocurrency-statistics/ (acesso em 27/10/2023).

[7] *DefiLlama*. URL: https://defillama.com/categories (acesso em 06/11/2023).

[8] *Ethereum Whitepaper*. URL: https://ethereum.org (acesso em 02/10/2023).

[9] Vincent Gramlich et al. “A Multivocal Literature Review of Decentralized Finance: Current Knowledge and Future Research Avenues”. Em: *Electronic Markets* 33.1 (abr. de 2023), p. 11. ISSN: 1422-8890. DOI: 10.1007/s12525-023-00637-4. URL: https://doi.org/10.1007/s12525-023-00637-4 (acesso em 09/11/2023).

[10] Farah Hasin et al. “ADS-B Based Air Traffic Management System Using Ethereum Blockchain Technology”. Em: *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*. Fev. de 2021, pp. 346–350. DOI: 10.1109/ICICT4SD50815.2021.9396828. URL: https://ieeexplore.ieee.org/document/9396828 (acesso em 19/11/2023).

[11] *Here’s How Much Was Lost to Crypto Hacks and Exploits in Q1 2023 | Bitcoin Insider*. URL: https://www.bitcoininsider.org/article/211488/heres-how-much-was-lost-crypto-hacks-and-exploits-q1-2023 (acesso em 01/10/2023).

[12] Rahmeh Fawaz Ibrahim, Qasem Abu Al-Haija e Ashraf Ahmad. “DDoS Attack Prevention for Internet of Thing Devices Using Ethereum Blockchain Technology”. Em: *Sensors* 22.18 (jan. de 2022), p. 6806. ISSN: 1424-8220. DOI: 10.3390/s22186806. URL: https://www.mdpi.com/1424-8220/22/18/6806 (acesso em 19/11/2023).

[13] *Judging Criteria*. URL: https://docs.code4rena.com/awarding/judging-criteria%5C#estimating-risk (acesso em 02/11/2023).

[14] *KEVM: A Complete Formal Semantics of the Ethereum Virtual Machine | IEEE Conference Publication | IEEE Xplore*. URL: https://ieeexplore.ieee.org/document/8429306 (acesso em 19/11/2023).

[15] Satoshi Nakamoto. “Bitcoin: A Peer-to-Peer Electronic Cash System”. Em: ().

[16] Ch Rupa et al. “Industry 5.0: Ethereum Blockchain Technology Based DApp Smart Contract”. Em: *Mathematical Biosciences and Engineering* 18.mbe-18-05-349 (2021), pp. 7010–7027. ISSN: 1551-0018. DOI: 10.3934/mbe.2021349. URL: http:

//www.aimspress.com/rticle/doi/10.3934/mbe.2021349 (acesso em 19/11/2023).

[17] Fabian Schär. *Decentralized Finance: On Blockchain- and Smart Contract-Based Financial Markets*. DOI: 10.20955/r.103.153-74. URL: https://research.stlouisfed.org/publications/review/2021/02/05/decentralized-finance-on-blockchain-and-smart-contract-based-financial-markets (acesso em 09/11/2023).

[18] *Sherlock*. URL: https://www.sherlock.xyz/ (acesso em 01/11/2023).

[19] *Smart Contracts Market Size, Share, & Trends [2023 Report]*. URL: https://www.grandviewresearch.com/industry-analysis/smart-contracts-market-report (acesso em 02/10/2023).

[20] *Solidity — Solidity 0.8.22 Documentation*. URL: https://docs.soliditylang.org/en/v0.8.22/ (acesso em 02/11/2023).

[21] *Solodit_content/Report_tags.Md at Main · Solodit/Solodit_content*. URL: https://github.com/solodit/solodit%5C_content/blob/main/report%5C_tags.md (acesso em 06/11/2023).

[22] *Technology Tipping Points and Societal Impact*. URL: https://www3.weforum.org/docs/WEF%5C_GAC15%5C_Technological%5C_Tipping%5C_Points%5C_report%5C_2015.pdf (acesso em 06/10/2023).

[23] Johnny Time. *The Most Interesting Web3 Security Interview with Peter Kacherginsky*. Out. de 2023. URL: https://medium.com/@JohnnyTime/the-most-interesting-web3-security-interview-with-peter-kacherginsky-cc03b0a30930 (acesso em 13/10/2023).

[24] "Turing Completeness". Em: *Wikipedia* (nov. de 2023). URL: https://en.wikipedia.org/w/index.php?title=Turing%5C_completeness%5C&oldid=1183169013 (acesso em 13/11/2023).

[25] Luyao Zhang, Xinshi Ma e Yulin Liu. *SoK: Blockchain Decentralization*. Ago. de 2023. arXiv: 2205.04256 [cs, econ, q-fin]. URL: http://arxiv.org/abs/2205.04256 (acesso em 19/11/2023).

[26] Zhuo Zhang et al. "Demystifying Exploitable Bugs in Smart Contracts". Em: *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. Melbourne, Australia: IEEE, mai. de 2023, pp. 615–627. ISBN: 978-1-66545-701-9. DOI: 10.1109/ICSE48619.2023.00061. URL: https://ieeexplore.ieee.org/document/10172700/ (acesso em 16/10/2023).