Uma análise classificatória em bugs encontrados em contratos inteligentes escritos em Solidity entre janeiro e setembro de 2023

Ana Julia Bittencourt Fogaça

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

1. Abstract

2. Resumo

Software é criado por seres humanos e, consequentemente, está sujeito a bugs. Isso não é diferente para aplicações que funcionam em blockchains que suportam a Ethereum Virtual Machine (EVM). Entretanto, em contraste com software convencional, as aplicações descentralizadas representam alvos particularmente lucrativos para hackers, dada a natureza transparente e de código aberto da blockchain. Neste artigo, analisamos 145 bugs descobertos durante auditorias públicas em vários projetos desenvolvidos em Solidity, operando na Ethereum ou em blockchains compatíveis com a EVM.

3. Introdução

Introduzida em 2008 pelo whitepaper do Bitcoin [11], a tecnologia blockchain é reconhecida como um vetor de transformação em múltiplas indústrias [17]. Suas características de segurança robusta, transparência e rastreabilidade, juntamente com a natureza de código aberto, contribuem para sua adoção em operações críticas de negócios. Até 2023, o uso mais evidente da blockchain, o mercado de criptomoedas, atingiu um valor de mercado superior a um trilhão de dólares [4]. A aplicabilidade da blockchain vai além das criptomoedas, influenciando áreas como finanças, gerenciamento de identidade, saúde e governança eleitoral [1].

A inovação do Ethereum, lançada pelo seu *whitepaper* em 2014, representou um marco na evolução da blockchain. Ao contrário do Bitcoin, concebido como uma moeda eletrônica *peer-to-peer* [11], o Ethereum apresentou o conceito inovador de "contratos inteligentes" [5]. Esta funcionalidade ampliou o escopo da blockchain, permitindo sua aplicação em novos domínios. A plataforma Ethereum se destaca pela hospedagem de uma máquina virtual que executa códigos em linguagens *Turing-complete*. No entanto, sendo os contratos inteligentes 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 atrativos para hackers. Apenas no primeiro trimestre de 2023, ataques à rede Ethereum resultaram no roubo de 320 milhões de dólares [7]. Para minimizar tais riscos, é prática comum realizar auditorias nos contratos inteligentes antes de seu lançamento em ambiente de produção. As auditorias podem ser privadas, conduzidas por empresas especializadas, ou públicas, através de plataformas como Code4rena [3] e Sherlock [13], onde participantes diversos podem identificar vulnerabilidades, sendo recompensados individualmente ou em grupo pela descoberta de bugs.

Desde 2020, projetos de blockchain que negligenciaram o processo de auditoria sofreram comprometimentos financeiros da ordem de 3.69 bilhões de dólares, enquanto que projetos auditados reportaram perdas de 1.3 bilhões[**CompetitiveVsPrivate**], sugerindo que, embora as auditorias reduzam a probabilidade de ataques bem-sucedidos, ainda há desafios na detecção antecipada de vulnerabilidades. Com a demanda por contratos inteligentes crescendo e uma projeção de aumento anual de 82,2% de 2023 a 2030 [14], é crucial entender e classificar as vulnerabilidades emergentes. Neste estudo, examinamos uma amostra de 145 bugs identificados em 31 competições de auditoria públicas realizadas entre janeiro e setembro de 2023 em duas plataformas de renome, Sherlock [13] e Code4rena [3]. Nosso objetivo é elucidar questões fundamentais como a complexidade na detecção de diferentes tipos de bugs, a incidência de categorias específicas de bugs em variados protocolos e a correlação entre vulnerabilidades exploradas por hackers e aquelas identificadas em competições de auditoria.

4. Revisão bibliográfica

4.1. Ethereum Blockchain

Ethereum, conforme delineado no whitepaper por Vitalik Buterin et al., é uma plataforma descentralizada que permite a construção de aplicações financeiras em cima de uma infraestrutura de blockchain[5]. A blockchain é um sistema de registro distribuído e imutável que mantém um registro contínuo de transações ou dados em blocos ligados por criptografia. Esse design assegura a integridade e a veracidade dos dados, resistindo a alterações retroativas.

4.2. Contratos inteligentes

Contratos inteligentes são programas que rodam na blockchain do Ethereum, permitindo que as partes cumpram acordos sem a necessidade de um intermediário. Uma vez implantados, os contratos inteligentes não podem ser alterados, o que exige que o código seja verificado para potenciais vulnerabilidades antes do lançamento. Eles são fundamentais para a finança descentralizada e têm bilhões de dólares em valor atrelados a eles[8].

4.3. Ethereum Virtual Machine

A Ethereum Virtual Machine (EVM) é o ambiente de execução para contratos inteligentes na Ethereum. Funciona como uma camada global que pode executar código de contrato inteligente em um contexto descentralizado[19]. Isso possibilita que os desenvolvedores criem aplicações que funcionam exatamente conforme programadas, sem qualquer possibilidade de fraude ou interferência de terceiros. A EVM é isolada, significando que o código executado dentro dela não tem acesso ao sistema de arquivos da rede, a outros contratos inteligentes, ou a qualquer recurso externo. Esse isolamento garante um alto nível de segurança no ecossistema Ethereum.

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[15]. 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 [**SolidityVulnerabilities**]. 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) representam uma infraestrutura financeira alternativa construída sobre a tecnologia blockchain, utilizando contratos inteligentes para replicar serviços financeiros existentes de forma mais aberta, interoperável e transparente[12]. DeFi é uma evolução tecnológica emergente que vem ganhando destaque juntamente com FinTech, RegTech, criptomoedas e ativos digitais, embora seu significado completo, implicações legais e consequências políticas ainda sejam pouco compreendidos. Este movimento revolucionário visa criar um sistema financeiro baseado apenas em código, sem intermediários, e viu um crescimento de ativos bloqueados de $4 bilhões para $104 bilhões em três anos[10]. Ainda é uma área complexa que exige uma compreensão rigorosa de suas nuances tanto por praticantes quanto por pesquisadores de sistemas de informação[6]. Como um novo setor de tecnologia financeira, DeFi tem o potencial de remodelar a estrutura da finança moderna e criar um novo cenário para empreendedorismo e inovação, prometendo e enfrentando desafios e limites[2].

5. Metodologia e perguntas da pesquisa

5.1. Coleta de dados

Nas 31 competições de auditoria públicas analisadas, que aconteceram de janeiro a setembro de 2023 nas plataformas Code4rena [3] e Sherlock [18] foram descobertos 145 bugs de alta severidade. Estas competições, com duração média de sete dias, visam a detecção de bugs críticos antes da implementação definitiva dos contratos inteligentes. O total de recompensas concedidas nestes eventos ultrapassou dois milhões de dólares, com uma média de 150 participantes por evento. Após a fase de submissão, juízes especializados em auditoria de contratos inteligentes, recrutados pela própria comunidade, avaliam a gravidade dos bugs encontrados. Aqueles categorizados como de alta severidade representam ameaças graves, incluindo a possibilidade de roubo ou perda de ativos digitais [9].

Plataforma	Categoria	Competição	Prêmio	HRF	nSLO
Code4rena	DAO	Arbitrum security council election system	90500	1	218
Code4rena	DAO	Llama	60500	2	209
Code4rena	Stablecoin	Lybra finance	60500	8	176
Code4rena	Dexes	Maia DAO ecosystem	300500	35	1099
Code4rena	Yield	PoolTogether	121650	9	332
Code4rena	Yield	PoolTogether v5: part deux	42000	2	100
Sherlock	Lending	Ajna update	85600	6	565
Sherlock	Yield Agreggator	Blueberry	72500	10	
Sherlock	Yield Agreggator	Blueberry Update #3	23600	5	363
Sherlock	Opções	Bond options	23600	2	88
Sherlock	Empréstimos	Cooler update	17000	4	51
Sherlock	Dexes	GFX labs	20400	2	71
Sherlock	Derivativos	GMX	200000	5	1057
Sherlock	Lending	Iron bank	67400	1	224
Sherlock	Derivativos	Perennial	122000	1	406
Sherlock	Derivativos	Perennial v2	125200	6	249
Sherlock	Derivativos	Symmetrical	91000	8	355
Sherlock	Derivativos	Symmetrical Update	27600	2	392
Sherlock	Launchpad	Tokensoft	21400	1	76
Sherlock	Stablecoin	Unitas protocol	36400	1	143
Code4rena	RWA	Ondo finance	60500	1	436
Sherlock	Índices	Index coop	130600	2	438
Sherlock	Stablecoin	USSD	18200	3	40
Sherlock	RWA	Dinari	16000	1	57
Sherlock	Dexes	RealWagmi	33200	5	108
Code4rena	DAO	Nouns DAO	100000	1	909
Sherlock	Dexes	DODO v3	57800	5	207
Sherlock	Derivativos	Hubble Exchange	60000	3	194
Code4rena	Stablecoin	Angle Protocol	52500	3	227
Code4rena	Gestores de liquidez	Arrakis	81400	2	280
Sherlock	Dexes	Unstoppable	36400	8	203
TOTALS			2255950	145	3095

5.2. Categoria dos protocolos

Os protocolos investigados neste estudo são dedicados ao setor de Finanças Descentralizadas (DeFi), abarcando exclusivamente as seguintes subcategorias conforme a classificação proposta por DefiLlama [**DefiLlama**]:

- 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.

5.3. Classificação dos Bugs

A classificação das vulnerabilidades dos protocolos de Finanças Descentralizadas (DeFi) analisados neste trabalho segue uma taxonomia híbrida, combinando os modelos propostos por Zhang et al. [20] e as tags de Solodit [16], 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 #TODO
  - Exploração baseado em *Flash loans* #TODO
- 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 #TODO
  - 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

5.4. Perguntas da pesquisa

- Q1: Que tipo de vulnerabilidade é mais difícil de ser encontrada por auditores?
- Q2: Que categoria de protocolo apresenta mais presença de bugs?
- Q3: Os auditores frequentemente perdem tipos específicos de bugs que são posteriormente explorados?
- Q4: Qual é o impacto financeiro médio de diferentes tipos de vulnerabilidades?
- Q5: Como a complexidade do contrato inteligente afeta a probabilidade de encontrar bugs?
- Q6: Qual a relação entre categoria de bugs e os diferente tipos de protocolos?

5.5. Dados coletados

5.6. Resultados

6. Referências

References

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

[2] Yan Chen and Cristiano Bellavitis. "Blockchain Disruption and Decentralized Finance: The Rise of Decentralized Business Models". In: *Journal of Business Venturing Insights* 13 (June 2020), e00151. ISSN: 2352-6734. DOI: 10.1016/j.jbvi.2019.e00151. URL: https://www.sciencedirect.com/science/article/pii/S2352673419300824 (visited on 11/09/2023).

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

[4] *Cryptocurrency Statistics 2023*. Sept. 2023. URL: https://www.forbes.com/advisor/au/investing/cryptocurrency/cryptocurrency-statistics/ (visited on 10/27/2023).

[5] *Ethereum Whitepaper*. URL: https://ethereum.org (visited on 10/02/2023).

[6] Vincent Gramlich et al. "A Multivocal Literature Review of Decentralized Finance: Current Knowledge and Future Research Avenues". In: *Electronic Markets* 33.1 (Apr. 2023), p. 11. ISSN: 1422-8890. DOI: 10.1007/s12525-023-00637-4. URL: https://doi.org/10.1007/s12525-023-00637-4 (visited on 11/09/2023).

[7] *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 (visited on 10/01/2023).

[8] *JCP | Free Full-Text | The State of Ethereum Smart Contracts Security: Vulnerabilities, Countermeasures, and Tool Support*. URL: https://www.mdpi.com/2624-800X/2/2/19 (visited on 11/02/2023).

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

[10] Eva Meyer, Isabell M. Welpe, and Philipp G. Sandner. *Decentralized Finance—A Systematic Literature Review and Research Directions*. SSRN Scholarly Paper. Rochester, NY, 2022. DOI: 10.2139/ssrn.4016497. URL: https://papers.ssrn.com/abstract=4016497 (visited on 11/09/2023).

[11] Satoshi Nakamoto. "Bitcoin: A Peer-to-Peer Electronic Cash System". In: ().

[12] 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 (visited on 11/09/2023).

[13] *Sherlock*. URL: https://www.sherlock.xyz/ (visited on 11/01/2023).

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

[15] *Solidity — Solidity 0.8.22 Documentation*. URL: https://docs.soliditylang.org/en/v0.8.22/ (visited on 11/02/2023).

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

[17] *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 (visited on 10/06/2023).

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

[19] Dr Gavin Wood. "ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER". In: ().

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