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

3. Introdução

Introduzida pela primeira vez em 2008 através do whitepaper do Bitcoin[8], a tecnologia blockchain é reconhecida como uma megatendência em computação com o potencial para transformar diversas indústrias[13]. Suas características únicas de segurança, transparência e rastreabilidade têm incentivado uma ampla variedade de setores a adotá-la para remodelar suas operações essenciais. Até 2023, o valor de mercado das criptomoedas, o caso de uso mais popular da blockchain até o momento, ultrapassou um trilhão de dólares[3]. A aplicabilidade da blockchain vai além das criptomoedas, abrangendo áreas como finanças, gerenciamento de identidade, saúde e governança eleitoral[1].

A publicação do whitepaper do Ethereum em 2014 marcou um avanço significativo na evolução da blockchain. Diferente do Bitcoin, que foi projetado inicialmente como uma versão p2p de dinheiro eletrônico[8], o Ethereum introduziu a noção revolucionária de contratos inteligentes[4]. Essa funcionalidade expandiu o alcance da tecnologia blockchain para novos setores. A inovação do Ethereum reside em sua capacidade de suportar uma máquina virtual que pode executar códigos em linguagens de programação *Turing-complete*. No entanto, como qualquer software, contratos inteligentes são desenvolvidos por humanos e, portanto, estão sujeitos a erros. Em um ambiente de código aberto - característica que é inerente à blockchains como o Ethereum, essas vulnerabilidades se tornam alvos lucrativos para hackers. Somente no primeiro trimestre de 2023, o Ethereum perdeu 320 milhões de dólares devido a ataques cibernéticos[5]. Para mitigar esses riscos, projetos que rodam em blockchains realizam auditorias antes de fazer o *deploy* da aplicação. Existem dos tipos de auditorias, as privadas, onde se é contratado uma empresa especialista em auditoria de contratos inteligentes, e as públicas, realizadas através de plataformas como a[9] e[2] onde qualquer individuo, empresa ou instituição pode participar e a recompensa é ofertada individualmente para o primeiro participante que achar ou é dividida entre todos os participantes que acharam o mesmo bug.

#TODO mencionar pesquisas ja realizados Com a crescente demanda por contratos inteligentes e uma expectativa de aumento anual de 82,2% de 2023 a 2030[10], torna-se fundamental compreender e categorizar as vulnerabilidades recentes. Neste artigo, analisamos um conjunto de dados de 154 bugs extraídos de 31 competições realizadas entre janeiro a setembro de 2023, através de duas plataformas de alta reputação,[9] e[2]. Nosso estudo busca esclarecer questões críticas, como a dificuldade de detecção de

diferentes tipos de bugs, a prevalência de certas categorias de bugs em distintos protocolos, e a relação entre as recentes vulnerabilidades exploradas por hackers e as vulnerabilidades encontradas nas competições.

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[4]. 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[6].

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[15]. 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[11]. 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

1. **TODO** explain DeFi

4.6. ERCs

5. Metodologia e perguntas da pesquisa

5.1. Coleta de dados

Foram identificados 145 bugs de alta severidade em 31 competições realizadas entre janeiro e setembro de 2023, nas plataformas Code4rena[2] e Sherlock [14]. As competições geralmente têm duração aproximada de 7 dias e têm como objetivo a identificação de bugs antes do deploy oficial. O montante total das recompensas distribuídas nas 31 competições ultrapassa a soma de dois milhões de dólares e contam, em média, com 150 participantes por competição. Após a etapa de participação, juízes — auditores de contratos inteligentes experientes e selecionados pela comunidade — determinam o nível de severidade dos bugs identificados. Bugs classificados como de alta severidade estão associados a riscos significativos, como o roubo ou a perda de ativos digitais [7].

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 analisados são exclusivamente do setor de Finanças Descentralizadas (DeFi) e englobam as subcategorias a seguir, seguindo a taxonomia de DefiLlama:

- Derivativos: Protocolos que oferecem instrumentos para negociações com alavancagem, permitindo aos usuários especular sobre os preços futuros de ativos e potencializar seus ganhos ou perdas.
- *Yield Farming*: Protocolos que incentivam os usuários a participar do staking ou a prover liquidez, recompensando-os por essas ações.
- Agregadores de *Yield*: Protocolos que maximizam os retornos ao combinar diferentes estratégias de *yield farming*.
- Opções: Protocolos que proporcionam o direito, mas não a obrigação, de comprar um ativo a um preço predeterminado em uma data futura.

- DAOs (Organizações Autônomas Descentralizadas): Estruturas organizacionais emergentes que operam sem uma entidade centralizada, com decisões tomadas coletivamente pelos participantes.
- Launchpads: Protocolos destinados a introduzir novos projetos e criptomoedas no mercado.
- Índices: Protocolos que acompanham ou replicam o desempenho de um conjunto de ativos correlacionados.
- DEXs (Trocas Descentralizadas): Protocolos que facilitam a troca de criptomoedas de maneira descentralizada.
- RWAs (Ativos do Mundo Real): Protocolos que envolvem a tokenização de ativos tangíveis, como propriedades imobiliárias.
- Stablecoins: Moedas digitais cujo valor é vinculado a moedas fiduciárias ou outros ativos, mantendo estabilidade por meio de mecanismos descentralizados.
- Gestores de Liquidez: Protocolos que administram posições de liquidez em market makers automatizados com liquidez focalizada.
- Empréstimos: Protocolos que facilitam o empréstimo e o empréstimo de uma variedade de ativos.

5.3. Classificação dos Bugs

A taxonomia utilizada foi uma junção da taxonomia apresentada por [16] e as tags apresentadas por Solodit[12].

- O: Fora do Escopo
  - Não é possível acessar o código-fonte do projeto.
  - Bugs que ocorrem em componentes fora da cadeia (off-chain).
  - Contratos inteligentes escritos em outra linguagem.
- C01: Manipulação do Mempool / Vulnerabilidades de Front-Running
  - Ataques sandwich #TODO
  - Flash loan exploits #TODO
- C02: Ataque de Reentrada Vulnerabilidades de reentrância ocorrem quando chamadas a contratos externos são feitas antes de atualizações de estado internas, permitindo que um adversário chame recursivamente o contrato, explorando o estado inconsistente.
- C03: Atualizações de Estado Errôneas. Ausência de atualização de estado ou atualizações incorretas, por exemplo, uma atualização de estado que não deveria estar presente.
- C04: *Hardcoded* configuração Prática de incorporar valores ou parâmetros fixos diretamente no código-fonte de um contrato inteligente. Isso pode representar um risco à segurança se a configuração precisar ser dinâmica ou adaptável.
- C05: Escalada de Privilégios e Problemas de Controle de Acesso.
  - Funções privilegiadas que podem ser chamadas por qualquer um ou a qualquer momento.
  - Fundos de usuários que podem ficar presos devido a código de retirada ausente ou incorreto.
- C06: Matemática Incorreta / Contabilidade Errônea. Matemática Incorreta refere-se a um problema potencial onde operações matemáticas dentro de um contrato inteligente são implementadas incorretamente, levando a cálculos imprecisos. Entre eles incluem:

  - Ordem de cálculo incorreta.
  - Retorno de um valor inesperado que desvia da semântica esperada especificada para o contrato.
  - Cálculos realizados com números incorretos (por exemplo, x = a + b ==> x = a + c, precisões incorretas).
  - Outros erros de contabilidade (por exemplo, x = a + b ==> x = a - b).
  - Underflow/overflow.
- C07: Lógica de Negócios Quebrada. Envolvem falhas na lógica de negócios ou protocolos de um contrato inteligente, onde a implementação corresponde à intenção do desenvolvedor, mas a lógica subjacente é inerentemente falha.
  - Sequências de invocação de função inesperadas ou ausentes (por exemplo, chamadas externas a contratos dependentes, sequências exploráveis que levam à realocação ou manipulação maliciosa de fundos).
  - Condições inesperadas do ambiente ou contrato
  - Argumentos de função inesperados.
- C08: Bugs Específicos da Implementação do Contrato. Estes bugs são difíceis de categorizar em outras categorias.
- C09: Falta de Proteção Contra Replay de Assinatura
  - Nonce ausente #TODO
  - Colisão de hash.
- C10: Verificação Ausente. Falha crítica no código de um contrato inteligente onde uma condição ou validação necessária não é implementada adequadamente.
- C11: DoS (Negação de Serviço). Vulnerabilidades de DoS ocorrem quando um atacante pode explorar um contrato de maneira que o torne irresponsivo ou significativamente menos eficiente. 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 Vulnerabilidades de validação de dados surgem quando um contrato inteligente não verifica ou saneia adequadamente as entradas, especialmente aquelas provenientes de fontes não confiáveis. Esta falta de validação pode levar a consequências não intencionais e potencialmente prejudiciais nas operações do contrato.
- C13: Correspondência de Lista Branca/Lista Negra. O contrato inteligente lida inadequadamente com endereços baseados em listas predefinidas.
- C14: Arrays. Vulnerabilidades relacionadas a arrays podem surgir quando os desenvolvedores não manuseiam adequadamente os índices de arrays ou falham em validar as entradas dos usuários.
  - leituras/escritas fora dos limites
  - problemas na exclusão
  - problemas com o 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] *Code4rena | Keeping High Severity Bugs out of Production*. URL: https://code4rena.com/ (visited on 11/01/2023).

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

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

[5] *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).

[6] *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).

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

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

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

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

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

[12] *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).

[13] *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).

[14] 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).

[15] Dr Gavin Wood. “ETHEREUM: A SECURE DECENTRALISED GENERALISED TRANSACTION LEDGER”. In: ().

[16] 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).