Análise de Ataques Sanduíche sob as Transações da Blockchain Ethereum

Alexandre Fontinele[1], Josué N. Campos[2], Isdael R. Oliveira[1], Glauber D. Gonçalves[1], José A. M. Nacif[2], Alex B. Vieira[3] e André C. B. Soares[1]

[1]Universidade Federal do Piauí (UFPI) – Picos, PI – Brasil
[2]Universidade Federal de Viçosa (UFV) – Florestal, MG – Brasil
[3]Universidade Federal de Juiz de Fora (UFJF) – Juiz de Fora, MG – Brasil

{isdael, alexandre, ggoncalves, andre.soares}@ufpi.edu.br

{josue.campos, jnacif}@ufv.br, alex.borges@ufjf.edu.br

Abstract. *The support for smart contracts on Ethereum has led to the emergence of a new decentralized and automated finance ecosystem called DeFi. This environment is highly competitive, and its protocols have been explored in search of vulnerabilities that offer economic profits to strategic users. Ethereum's pending transaction pool has recently become the target of financial speculation. In search of making some profit, attackers continuously monitor the pool and try to anticipate transactions from other users. They strategically insert their transactions before and after the potential victim's transaction, known as a sandwich attack. This paper evaluates potential sandwich attacks on Ethereum over 2023, updating knowledge about this approach. Our studies based on 113,774 of 2,599,105 blocks show 1,553,362 possible attacks, with an average profit of USD 3,202.82 for attackers, providing strong evidence that sandwich attacks continue to occur in the DeFi ecosystem.*

Resumo. *O suporte a contratos inteligentes na Blockchain Ethereum propiciou a emergência de um novo ecossistema de finanças descentralizado e automatizado, denominado DeFi. Esse ambiente é altamente competitivo e seus protocolos vem sendo explorados em busca de vulnerabilidades que oferecem ganhos econômicos a usuários estratégicos. Recentemente, a fila de transações pendentes do Ethereum tornou-se alvo de especulações financeiras. Na esperança de obter algum lucro, atacantes monitoram continuamente a fila e tentam antecipar transações de outros usuários, inserindo estrategicamente suas transações antes e após a transação da potencial vítima, o que se tornou conhecido como ataque sanduíche. Neste artigo, avaliamos suspeitas de ataques sanduíche na Blockchain Ethereum durante o ano de 2023, atualizando os conhecimentos sobre esse ataque. Nossas análises baseadas em 113.774 dos 2.599.105 blocos processados demonstram a ocorrência de 1.553.362 especulações de ataques, com um lucro de em média USD 3.202,82 para os atacantes, fornecendo fortes evidências que ataques sanduíche continuam ocorrendo no ecossistema DeFi.*

1. Introdução

Ethereum é uma plataforma popular baseada na tecnologia blockchain para negociação de ativos digitais [Chen et al. 2020]. Desde o seu início em 2014, o Ethereum registrou quase

2 bilhões de transações e tem mais de R$1,3 trilhões em valor de mercado, tornando-se a segunda maior plataforma atualmente.[1] A rede Ethereum estende o Bitcoin, primeira plataforma blockchain, permitindo o controle de ativos digitais por meio de códigos de programas denominados contratos inteligentes. Ao fornecer ao usuário diversas possibilidades para determinar como utilizar e transferir seus ativos, os contratos inteligentes contribuem enormemente para aplicações disruptivas envolvendo blockchains para produtores e consumidores de serviços financeiros, colecionáveis, entre outros. [Xu et al. 2019].

Por sua vez, DeFi (i.e, *Decentralized Finance*) é um ecossistema de contratos inteligentes que visam replicar, aprimorar ou substituir os serviços financeiros tradicionais por meio de negociação e transferência de objetos digitais ou *tokens* em blockchain [Harvey et al. 2021]. A ideia de DeFi é eliminar intermediários e barreiras para o crédito no sistema bancário vigente, fornecendo acesso mais amplo a serviços financeiros no contexto da web descentralizada. Nesse sentido, DeFi oferece operações financeiras fundamentais (câmbio, saque e empréstimo com ou sem garantias) na forma de protocolos codificados em contratos de blockchains populares como Ethereum, Binance, Avalanche dentre outras. Esses protocolos permitem que usuários interajam diretamente com a blockchain, e permitem também que desenvolvedores e usuários combinem diferentes protocolos para criar soluções financeiras personalizadas e inovadoras.

Uma das principais ameaças ao ecossistema DeFi no Ethereum atualmente é a manipulação na ordem de transações que serão efetivadas na blockchain por usuários estratégicos que monitoram constantemente a fila pública de transações pendentes (i.e., *mempool*), ataque esse conhecido como *front-running*. Nesse tipo de ataque, o valor da tarifa da transação é utilizado pelo atacante para manipular a ordem das transações que aguardam na *mempool* para constituírem o novo bloco. Por essência, o consenso distribuído em blockchain elimina uma autoridade central para gerenciar a *mempool* e evitar ataques *front-running*. Assim, vem ocorrendo um aumento no número de atacantes, bem como várias formas de ataques *front-running* [Varun et al. 2022, Torres et al. 2021, Zhang et al. 2023b, Zhang et al. 2023a].

O ataque sanduíche, que é um tipo *front-running*, é uma estratégia de negociação já conhecida nos sistemas financeiros tradicionais, onde um usuário com visão privilegiada do sistema identifica um negócio promissor de outro usuário e o executa antecipadamente obtendo os benefícios desse negócio. Esse tipo de ataque vem chamando a atenção de pesquisadores recentemente no contexto da blockchain Ethereum e DeFi [Torres et al. 2021, Zhang et al. 2023a]. Nesse caso, um atacante inicia monitorando a *mempool* em busca de transações pendentes que estão prestes a negociar grandes somas de um determinado ativo. Uma grande transação resultará em uma flutuação no preço do ativo. Na sequência, o atacante cria o chamado "sanduíche", cercando esta grande transação com duas de suas próprias transações. Na primeira transação, o atacante executa uma grande transação para comprar ou vender alguma quantidade de ativo antes que o preço do ativo flutue. Na segunda transação, o atacante retrocede a grande transação para recomprar o ativo original por um preço mais baixo ou vender o ativo recém-adquirido por um preço mais alto. Em ambos os casos o atacante obtém lucro devido à diferença de preço e vítima pode sofrer prejuízo [Weintraub et al. 2022].

[1] `https://coinmarketcap.com/pt-br/currencies/ethereum/`

Estudos recentes mostram a prevalência e gravidade de ataques *front-running* através de medição desses ataques na Blockchain Ethereum [Eskandari et al. 2020, Daian et al. 2019, Torres et al. 2021, Varun et al. 2022, Zhang et al. 2023a]. Os estudos em [Daian et al. 2019] apontam que os ataques *front-running* representam uma grande ameaça ao ecossistema do blockchain. Em [Eskandari et al. 2020], os autores conduziram um estudo de caso sobre quatro categorias de contratos inteligentes e identificaram três padrões de ataques: deslocamento, inserção (ataque sanduíche) e supressão. Os autores em [Torres et al. 2021] descobriram que os ataques *front-running* são predominantes na Blockchain Ethereum e causaram uma perda total de mais de 18,41 milhões de dólares. Em [Varun et al. 2022] é apresentado um forma de detecção e prevenção de ataques *front-running* baseado em modelo. Enquanto que em [Zhang et al. 2023a] é proposto um algoritmo para identificação de ataques *front-running* e realizado um estudo para verificar a eficiência de outras propostas de identificação de ataques *front-running*. Os estudos realizados em [Torres et al. 2021, Zhang et al. 2023a] mostram que a quantidade de ataques sanduíches na Blockchain Ethereum é muito superior a quantidade dos outros ataques *front-running*. Em nenhum desses trabalhos são analisadas tentativas mais recentes de ataques sanduíche se limitando a medições anteriores a 2022. Portanto há uma falta de informações atuais sobre a incidência desse ataque na rede Ethereum, assim como até que ponto chegam os custos e lucros dos atacantes com a sua aplicação.

Neste artigo, analisamos ocorrências de ataques sanduíche na Blockchain Ethereum no ano de 2023, atualizando os conhecimentos sobre esse ataque. O objetivo principal é avaliar como essa ameaça ao ecossistema DeFi vem se evoluindo, visto a falta de medições mais recentes. Para isso, foi desenvolvido um arcabouço para coleta e processamento de blocos via vários computadores paralelamente. Foram processados um total de 2.599.105 de blocos criados em 2023, inspecionando em cada bloco tuplas de transações (i.e., atacante-vítima-atacante) com características de ataques sanduíches mais abrangentes às da literatura acima mencionadas. Especificamente, observamos exemplos mais recentes de ataques sanduíches e atualizamos essas características para incluir tentativas menos convencionais de ataques, onde os atacantes adotam diferentes estratégias para lidar com o mecanismo de tarifação do Ethereum e manipular a ordem de transações no bloco. Logo, é importante analisar quão bem sucedidos tais tentativas são em comparação aos ataques cujas características já são mapeadas na literatura.

Nossos resultados indicam que os ataques com características mais abrangentes são predominantemente recorrentes na rede. Identificamos mais de 1,5 milhões de tentativas de ataques cujas estratégias de pagamento de tarifas para manipular ordens de transação não são claras, ao passo que apenas 3.774 tentativas seguem a estratégia mais restrita [Torres et al. 2021, Zhang et al. 2023a] foram encontradas. Sendo assim, organizamos nossa análise sob duas perspectivas de ataque: (1) a perspectiva abrangente e (2) a perspectiva restrita. Na primeira medimos ataques mais ousados que levaram a lucros exorbitantes de até USD 162 milhões (média de USD 3,2 mil por ataque), mas com prejuízos para cerca de 50% das tentativas de ataques, dado que os custos com tarifações e aquisição do ativo superaram o lucro com a sua venda. Por sua vez, observamos ataques moderados em nossa análise sob a perspectiva restrita, i.e., uma lucratividade menor com média em torno de USD 1,8 mil por ataque, mas com mais de 75% das tentativas de ataques alcançando lucro. Apesar dos riscos associados aos ataques características abrangentes, verificamos a sua tendência de crescimento mensal a partir de julho de 2023

superando notavelmente a quantidade de ataques restritos acima de 5.000 tentativas.

As próximas seções deste artigo tem a seguinte organização. Na Seção 2 apresentamos uma visão geral sobre ataques sanduíche e na Subseção 2.3 apresentamos a metodologia e o algoritmo utilizado para a coleta dos dados. A Seção 3 apresenta com mais detalhes os resultados obtidos, demonstrando o método utilizado para o agrupamento e caracterização dos dados. Na Seção 4, discutimos os trabalhos relacionados e, finalmente, a Seção 5 expõe as considerações finais e trabalhos futuros.

2. Ataque Sanduíche

Esta seção apresenta a definição dos ataques sanduíche e como eles ocorrem nas redes Blockchains, assim como a metodologia adotada para identificação das suspeitas desses ataques.

2.1. Visão Geral

Qualquer ação realizada pelo usuário que modifique o estado da blockchain é registrada como uma transação. Uma vez efetivada, a transação não pode ser revertida e é esse recurso que torna a blockchain um livro razão imutável. Nesse sentido, os contratos inteligentes, assim como os usuários externos, são capazes de armazenar ativos em contas endereçáveis e realizar transações que alteram o estado da rede [Varun et al. 2022].

Contratos inteligentes são programas criados usando uma linguagem de programação de alto nível como Solidity. Eles são projetados para serem executados quando um conjunto predeterminado de condições forem atendidas. Geralmente, eles automatizam a execução de um ativo ou acordo que garante que todos os participantes conheçam instantaneamente o resultado, sem o envolvimento de qualquer parte intermediária. Os contratos inteligentes também podem automatizar processos, ou seja, desencadear outra ação assim que o contrato atual for executado. Uma vez que são implantados na blockchain, quando as condições exigidas são atendidas, eles são executados por uma rede de nós que chegam a um consenso antes que o estado executado seja armazenado na blockchain.

Uma blockchain é um registro de transações anexado. Por sua vez, as transações são armazenadas em uma *pool* e tratadas igualmente, independentemente do horário específico em que elas foram adicionadas, para posteriormente serem combinadas em um bloco por nós mineradores. Dessa maneira, mais de 95% dos mineradores optam por ordenar as transações em relação ao preço do gás, que é a taxa de transação que os mineradores recebem por anexar uma transação em um bloco.

Os atacantes podem ser mineradores ou não mineradores. Os mineradores não são obrigados a pagar um preço mais alto de gás para manipular a ordem das transações, pois têm controle total sobre as transações que são incluídas em um bloco. Os não mineradores, por outro lado, são obrigados a pagar um preço mais elevado de gás para antecipar as transações de outros não mineradores. Assim como em [Torres et al. 2021], nós assumimos que o atacante é um não minerador financeiramente racional com a capacidade de monitorar a *mempool* de transações. O atacante precisa processar as transações na *mempool*, encontrar uma vítima e criar transações de ataque antes que a transação da vítima seja minerada.

O atacante não seria capaz de reagir rápido o suficiente para realizar todas as tarefas necessárias para efetuar o ataque manualmente. Portanto, seguindo [Torres et al. 2021], assumimos que o atacante possui pelo menos um programa de computador (*Bot*) que executa automaticamente as tarefas necessárias para o ataque. O *Bot* precisa de pelo menos uma ou mais contas de propriedade externa (EOA - *Externally Owned Accounts*) para atuar como remetente de qualquer transação de ataque. O uso de várias contas de propriedade externa auxilia os atacantes a ocultar suas atividades, semelhante aos esquemas de lavagem de dinheiro. Assumimos que o atacante possui um saldo suficientemente grande em todas as suas contas, a partir do qual pode enviar transações de ataque com gás suficientemente maior que o gás da transação da vítima. No entanto, o atacante também pode empregar contratos inteligentes para manter parte da lógica do ataque. Esses contratos inteligentes são referidos como contratos de *Bot*, que são chamados pelas contas do atacante.

A Figura 1 apresenta um cenário com ataque sanduíche. No ataque sanduíche, o atacante envia duas transações, uma com um preço de gás mais alto do que a transação da vítima e outra com um preço de gás mais baixo para intercalar a transação da vítima. É usado em plataformas descentralizadas de câmbio (DEXes - *Descentralized Exchanges*) para fazer sanduíches de transações prestes a negociar grandes somas de um determinado ativo, também conhecido na literatura como transações baleia [Varun et al. 2022].

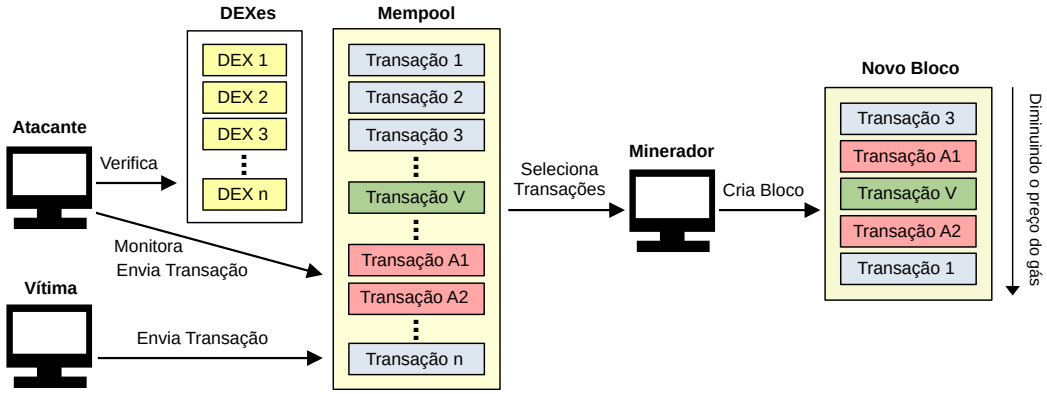

Figura 1. Demonstração de um cenário com ataque sanduíche.

Observa-se na Figura 1 que o atacante monitora a *mempool* na espera de uma possível transação vítima (Transação V). O atacante verifica nas DEXes se o ativo da transação da vítima pode gerar lucro. Em caso positivo, o atacante gera duas transações (Transação A1 e Transação A2). Elas, por sua vez, possuem valores de gás que fazem com que realizem um "sanduíche" com a transação da vítima quando o minerador selecionar transações para criar um bloco seguindo a ordenação convencional pelo preço do gás.

2.2. Plataformas Descentralizadas de Câmbio

Criadores de Mercado Automatizados (AMM – *Automated Market Makers*) é um modelo de protocolo utilizado por DEXes que tem por objetivo definir a precificação de ativos digitais. O protocolo AMM se baseia em uma fórmula matemática para determinar os preços dos ativos. Alguns AMMs, inclusive, usam fórmulas simples, como é o caso do Uniswap. Provedores de liquidez são usuários que depositam ativos em *pools* de liquidez,

que são espaços controlados por contratos inteligentes, e em troca recebem recompensas. Qualquer ordem única de compra ou venda pode ser executada independentemente de outras negociações em AMM DEXes. Quando os comerciantes desejam trocar os ativos, eles invocam funções de contratos inteligentes que os transferem entre a *pool* de liquidez e a conta dos comerciantes. A taxa de câmbio entre estes ativos é determinada por funções predefinidas de forma transparente codificadas no contrato inteligente da AMM DEX [Qin et al. 2021, Wang et al. 2022].

Adicionalmente, uma ordem de compra aumentará o preço de um ativo, enquanto uma ordem de venda diminuirá o preço do ativo. Portanto, os especuladores podem monitorar continuamente a rede (*i.e. mempool* e AMMs) para encontrar transações pendentes para AMM DEXes (*i.e.* transação da vítima) que implicarão diferenças de preços. Dessas DEXes, os especuladores podem comprar o ativo por um preço baixo antes que a transação da vítima seja executada (transação T_{A1}) e venderem o ativo após a transação da vítima aumentar de preço (transação T_{A2}), gerando lucro para si. Ao final do ataque, a cotação da transação vítima é pior do que seria sem a transação T_{A1}, resultando em perda financeira para a vítima [Wang et al. 2022].

2.3. Algoritmo de Coleta de Dados

O algoritmo de detecção de arbitragem de ataques sanduíche utiliza o fato de que as DEXes fazem uso do padrão de token ERC-20. O padrão ERC-20 define muitas funções e eventos que permitem aos usuários negociar seus tokens entre si e entre corretoras. Em particular, sempre que um token é negociado, um evento chamado *Transfer* é acionado e informações sobre o remetente, o destinatário e o valor são armazenados na blockchain. Combinando essas informações com dados transacionais (por exemplo, ordem de transações, preço do gás, etc.) é possível detectar a ocorrência de arbitragem de ataques sanduíche [Torres et al. 2021].

Para identificar uma arbitragem de ataque sanduíche, são verificadas todas as transações de transferência em um bloco. Uma transação de transferência é definida como $T = (s, r, a, c, h, i)$, em que s é remetente dos tokens, r é o receptor dos tokens, a é o número de tokens transferidos, c é o endereço do contrato do token, h é o hash da transação, e i é o índice da transação. Ao verificar todas as transações em um bloco, busca-se encontrar três transações de transferência: T_{A1}, T_V e T_{A2}. As transações T_{A1} e T_{A2} estão relacionadas com o atacante, e a transação T_V está relacionada com a vítima.

O algoritmo para detecção de arbitragem de ataque sanduíche segue as suposições a seguir. As transações T_{A1}, T_V e T_{A2} devem estar nessa ordem, ou seja, o índice de T_{A1} deve ser menor do que o índice de T_V e o índice de T_V deve ser menor que o índice de T_{A2} ($i_{A1} < i_V < i_{A2}$). O atacante e a vítima compram tokens em T_{A1} e em T_V, respectivamente. Em seguida, o atacante vende os tokens em T_{A2} que comprou anteriormente em T_{A1}. O número de tokens comprados por T_{A1} deve ser semelhante ao número de tokens vendidos por T_{A2} (ou seja, $a_{A1} \approx a_{A2}$). O atacante e a vítima realizam transações sobre os mesmos tokens, ou seja, os endereços de contrato de token de T_{A1}, T_V e T_{A2} devem ser idênticos ($c_{A1} = c_V = c_{A2}$). O remetente de T_{A1} deve ser idêntico ao remetente de T_V, bem como o receptor de T_{A2}, e o receptor de T_{A1} deve ser idêntico ao remetente de T_{A2} (ou seja, $s_{A1} = s_V = r_{A2} \wedge r_{A1} = s_{A2}$). Os *hashes* de transação de T_{A1}, T_V e T_{A2} devem ser diferentes (ou seja, $h_{A1} \neq h_V \neq h_{A2}$). Vale ressaltar que, as suposições e o algoritmo

desenvolvido foram baseados e adaptados com base no trabalho de [Torres et al. 2021], com o objetivo de suportar os novos dados de possíveis ataques coletados durante o ano de 2023, tanto no que diz respeito à quantidade de dados coletados quanto às novas abordagens dos ataques ocorrerem na rede.

O Algoritmo 1 foi utilizado para a coleta dos dados. A busca para identificar uma arbitragem começa por localizar a segunda transação do ataque T_{A2} (Linha 3). Depois tenta localizar a primeira transação T_{A1} (Linha 5). Na linha 7 é verificado se a diferença entre os tokens que foram comprados em T_{A1} e os tokens que foram vendidos em T_{A2} não é maior que 1%. Se a diferença não for maior que 1%, então busca identificar a transação vítima T_V (Linha 9). Na linha 11 verifica se as três transações (T_{A1}, T_V e T_{A2}) estão realizando transferências com a mesma DEX e com o mesmo token. Em caso de positivo, um ataque é identificado e guardado na lista de ataques A. Em seguida, continua com a busca para identificar possíveis outros ataques (Linha 14).

Algoritmo 1 : Coleta de ataques sanduíche em um bloco.

Require: B {Um Bloco com todas as suas transações.}
1: A {Para guardar os ataques sanduíche encontrados no Bloco.}
2: **for** $i_{A2} \leftarrow 3$ **to** $|B| - 1$ **do**
3: $T_{A2} \leftarrow getTxAtIndex(B, i_{A2})$
4: **for** $i_{A1} \leftarrow i_{A2} - 2$ **to** 0 **do**
5: $T_{A1} \leftarrow getTxAtIndex(B, i_{A1})$
6: **if** $T_{A1}[src] \neq T_{A2}[src]$ **then continue**
7: **if** $valueDifGreaterRate(T_{A1}, T_{A2}, 0.01)$ **then continue**
8: **for** $i_V \leftarrow i_{A1} + 1$ **to** $i_{A2} - 1$ **do**
9: $T_V \leftarrow getTxAtIndex(B, i_V)$
10: **if** $T_V[value] = 0$ **then continue**
11: **if** $sameDEX(T_{A1}, T_V, T_{A2})$ **and** $sameToken(T_{A1}, T_V, T_{A2})$ **then**
12: $a \leftarrow arbitrage(T_{A1}, T_V, T_{A2})$
13: $saveArbitrage(A, a)$ {Guarda o ataque identificado.}
14: $goTo(NEXT)$
15: **end if**
16: **end for**
17: **end for**
18: $NEXT$ {Marcação para a função $goTo()$.}
19: **end for**
20: **return** A

O Algoritmo 1 assume que os ataques sanduíche sempre ocorrem dentro do mesmo bloco. Essa suposição permite verificar os blocos em paralelo, uma vez que só é preciso comparar as transações dentro de um bloco. No entanto, esta suposição nem sempre se aplica à realidade, uma vez que as transações podem ser dispersas por diferentes blocos durante o processo de mineração. Sendo assim, podem existir ataques sanduíches realizados em vários blocos e que o algoritmo não é capaz de detectar. Portanto, a abordagem apresentada representa um limite inferior para análises de ataques sanduíches na blockchain Ethereum. Ela considera o compromisso entre analisar o ataque típico e o custo para essa análise. Em outras palavras, entendemos que o atacante busca o lucro

rápido concentrando seus esforços em um mesmo bloco (*i.e.*, ataque típico). Nesse sentido, o esforço computacional de estender a análise a sucessivos blocos identificaria, em tese, menos ataques. Logo, os resultados discutidos a seguir é uma linha base de alertas sobre a necessidade de lidar com essa ameça ao ecossistema DeFi, quando o volume real de ataques pode ser ainda maior. Contudo, existem abordagens que expandem a análise para mais de um bloco, como no trabalho de [Zhang et al. 2023a] que analisa os ataques em uma janela de 3 blocos consecutivos.

3. Resultados

A fim de identificar as suspeitas de ataques sanduíche ocorrentes na Blockchain Ethereum, coletamos dados de transações durante o período de janeiro à dezembro de 2023. Nesta seção descrevemos, primeiramente, informações detalhadas sobre a coleta de dados e, logo em seguida, apresentamos as análises resultantes desses dados.

3.1. Configuração Experimental

O algoritmo de identificação de suspeitas de ataques (Algoritmo 1) examina individualmente cada bloco do ano de 2023. Como o algoritmo analisa cada bloco de maneira individual, ele pode ser executado de forma simultânea em vários blocos. Isso foi possível com a implementação do algoritmo em um sistema com suporte a múltiplas *Threads*, onde cada *Thread* é atribuída a um bloco específico para análise.

Para que o algoritmo consiga identificar os ataques, é fundamental que haja acesso ao bloco completo, incluindo todas as suas transações. Nesse sentido, a integração das APIs do Quicknode e do Alchemy é fundamental para o algoritmo de detecção de ataques sanduíche na rede Ethereum. As informações coletadas pelas APIs dizem respeito a dados específicos da blockchain, como as transações ocorridas, endereços de origem e destino das transações e os valores transferidos, detalhes dos blocos como momento de criação e quantidade de transações existentes, bem como informações referentes aos contratos inteligentes, como endereço de localização na rede.

Em relação ao volume de requisições, para cada bloco é necessário uma requisição para a API até a composição de um ano de coleta de dados. Além disso, para cada ataque identificado é necessário realizar requisições com o objetivo de: confirmar os endereços de usuário, coletar o endereço do contrato utilizado, nome do *token* e o nome da DEX envolvida no ataque, por exemplo. Dessa maneira, foram utilizadas quatro máquinas virtuais da AWS (*Amazon Web Services*) trabalhando continuamente por um período de duas semanas para atingir tal objetivo. Quando o algoritmo detecta um ataque, as informações relevantes são armazenadas permanentemente em um servidor local MongoDB. Para cada transação, dados como endereço da conta de origem, custo de gás, corretora utilizada e *token* transferido foram armazenados, com o objetivo de serem processados posteriormente seguindo as suposições estabelecidas. Os *scripts* utilizados para a coleta, análise e os dados obtidos estão no repositório `https://github.com/LABPAAD/blockchain_defi`.

Por meio da configuração e metodologia apresentadas, identificamos um total de 1.553.362 suspeitas de ataques no ano de 2023 em 113.774 blocos dos 2.599.105 processados, ou seja, em alguns blocos foram identificadas mais de uma suspeita. Além disso, dessa quantidade, coletamos 2.103 contas de possíveis atacantes diferentes envolvidos,

assim como 568.444 possíveis vítimas afetadas. Vale mencionar que todos os casos de suspeitas de ataques analisados nesse trabalho podem ser verificados detalhadamente no repositório de dados disponibilizado.

3.2. Análises

A partir do processamento dos dados coletados, foi possível inferir as métricas de custos envolvidos, lucros obtidos pelos atacantes e as variações no preço do gás tendo em vista a primeira transação do atacante, a transação da vítima e a segunda transação do atacante. Calculamos o custo para cada suspeita de ataque como a soma da quantidade de Ether que um atacante gasta em sua primeira transação (T_{A1}) e as taxas dessa e da segunda transação (T_{A2}). O lucro de um ataque é calculado como a quantidade de Ether que um atacante ganha em T_{A2} menos o custo. Para cada ataque, convertemos o custo e o lucro em Ether para dólares americanos, tomando a taxa de conversão válida no momento do ataque. Adicionalmente, calculamos Δ_1 e o Δ_2, respectivamente, como o preço do gás de T_{A1} menos o preço do gás da transação da vítima (T_V) e o preço do gás de T_V menos o preço do gás de T_{A2}.

Tabela 1. Distribuições para os supostos ataques sanduíche em 2023 sob as Perspectivas Abrangente (PA) e Restrita (PR).

Distribuição	Custo (USD)		Lucro (USD)		Δ_1 Gás (GWei)		Δ_2 Gás (GWei)	
	PA	PR	PA	PR	PA	PR	PA	PR
Média	127,31	63,18	3.202,82	1.801,96	-3,90	65,52	-293,76	5,46
Desvio Padrão	345,36	95,17	212.322,54	4.494,06	42,98	189,94	1.178,79	16,02
Mínimo	1,20	2,07	-5.226.396,86	-12.015,46	-10.000,00	0,00	-255.217,48	0,00
25%	19,90	17,90	-31,82	2,39	-2,00	6,82	-262,22	0,10
50%	47,05	36,25	-0,58	50,58	-0,30	17,56	-92,77	0,30
75%	118,88	81,44	0,09	1.363,16	-0,10	54,42	-28,26	3,00
Máximo	41.530,91	1.803,26	162.884.574,94	30.226,80	15.553,69	4.029,20	9.180,83	450,00

A Tabela 1 mostra a distribuição acumulada para custos, lucros e variações de gás para os ataques coletados em 2023. Para uma compreensão mais ampla sobre ataques sanduíches com essas métricas utilizamos duas perspectivas de análise: (1) perspectivas abrangente (PA) e (2) perspectiva restrita (PR). A primeira considera todos os ataques identificados pelo Algoritmo 1 ao passo que a segunda considera dentre esses apenas os ataques com variações de gás positivas, ou seja, há uma coordenação dos valores de tarifas em T_{A1} e T_{A2}, o que caracteriza a intenção de um ataque segundo [Torres et al. 2021]. Em outras palavras, a perspectiva restrita reduz falsos positivos, pois considera como indícios da intenção do atacante em posicionar suas duas transações antes e após a transação da vítima pagando uma tarifa maior que a vítima em T_{A1} (*i.e.*, $\Delta_1 > 0$) e uma tarifa menor que a vítima em T_{A2} (*i.e.*, $\Delta_2 > 0$).

Na perspectiva abrangente mostrada na Tabela 1, os prejuízos chegam a 50% das observações, o que significa muitas tentativas mal sucedidas de ataques com custos superiores aos lucros. Além disso, o prejuízo de mais de 5 milhões indica casos extremos de atacantes perdendo dinheiro. Contudo, houve compensação para uma parcela dos ataques (75^o percentil), alcançando-se lucro máximo superior a USD 162 milhões. Por sua vez, a análise que considera um ataque sob a perspectiva restrita indica uma quantidade de 4.042 ataques e mostra uma maioria de tentativas com lucros (25^o percentil). Nesses casos, observa-se que o custo e o lucro embora não muito elevados para a maioria dos ataques têm distribuições de ambas as métricas com caudas longas para a direita, indicadas com faixas de custos entre USD 82-1.803 e lucros entre USD 1.370-30.226 para o

75^o percentil e o valor máximo. Apesar da ocorrência de prejuízos que alcançam até USD 12.015, eles representam menos de 25% das tentativas de ataques. Isso significa que há atacantes estratégicos que conseguem extrair oportunidades reais de ganhos financeiros com ataques sanduíche.

Como mostra a Figura 2, é notável o crescimento do lucro conforme o avanço dos meses no ano de 2023 para a perspectiva abrangente. Por exemplo, no mês de dezembro, o lucro obtido pelos atacantes foi cerca de USD 8.910,77. Já para a perspectiva restrita, observa-se um lucro com uma certa tendência de padrão na maioria dos meses, porém com baixas e altas, como em fevereiro e maio, respectivamente. Em ambos os casos, é possível notar que, na maioria dos meses em média os atacantes conseguem obter lucros por meio da prática desse ataque. Nos meses de abril e novembro, por exemplo, observa-se uma média de lucro aproximadamente iguais para ambas as perspectivas.

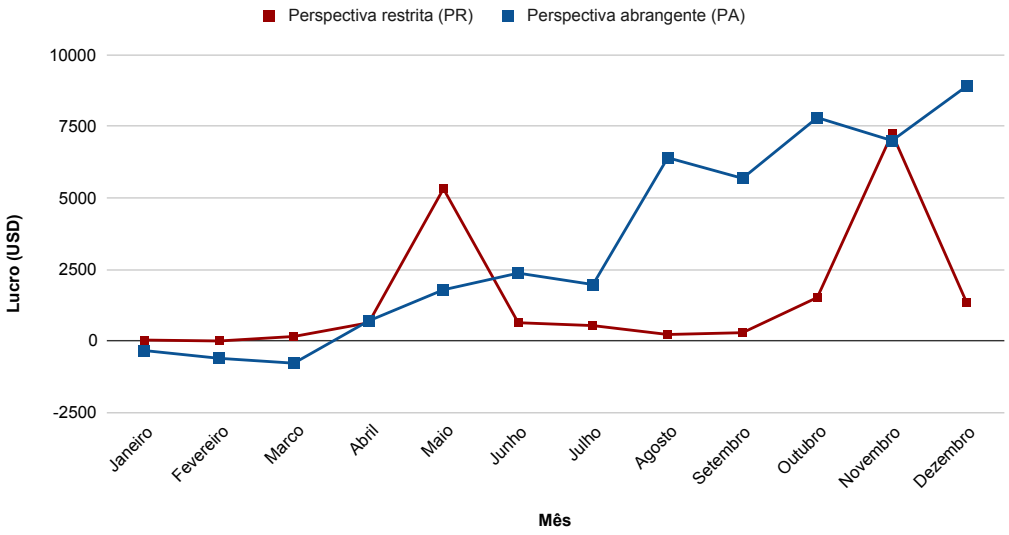

Figura 2. Crescimento do lucro médio dos atacantes por mês em 2023.

Por outro lado, por meio do processamento dos dados foi possível analisar os *tokens* que mais foram alvos das suspeitas de ataques sanduíche, assim como mostra a Tabela 2. Para a perspectiva abrangente, o Pepe Coin, criptomoeda deflacionária, foi alvo de 13.168 ataques, seguido pelo USD Coin, criptomoeda ligada ao dólar dos Estados Unidos, com 11.480 ataques. Por outro lado, além desses *tokens*, é possível notar um crescente aumento de ataques em *tokens* recém criados na rede Ethereum e *tokens* não conhecidos popularmente, de acordo com os dados coletados para a perspectiva restrita. Uma hipótese para que esses *tokens* novos sejam alvos de atacantes na perspectiva restrita é que esses *tokens* possuem menos liquidez, alta volatilidade e uma base menor de usuários, o que pode torná-los mais suscetíveis a manipulações de preço por agentes maliciosos.

Adicionalmente, foram coletadas informações acerca das corretoras mais utilizadas dentre os dados processados, assim como os *bots* mais utilizados. Estes *bots* possuem o objetivo de identificar transações que sejam lucrativas, levando em consideração aspectos como a tarifa oferecida para realizar a transação e o custo computacional, *i.e.* o valor e consumo do gás de determinada transação. Como mostra a Figura 3, a corretora Uniswap

Tabela 2. *Tokens* que mais foram alvos das suposições de ataques em 2023.

Perspectiva abrangente		Perspectiva restrita	
Token	**Qtde.**	**Token**	**Qtde.**
Pepe Coin	13.168	Raptor	145
USD Coin	11.480	Chooky	87
Tether USD	9.849	SMORT	56
HarryPotterObamaSonic10Inu	7.049	Nord Network	40

V2 possui a maior quantidade de relação com as suspeitas de ataques sanduíche analisados, com 109.833 ocorrências. Já na Tabela 3, o endereço do contrato *bot* com mais ocorrências possui 35.465 relações com as arbitragens coletadas.

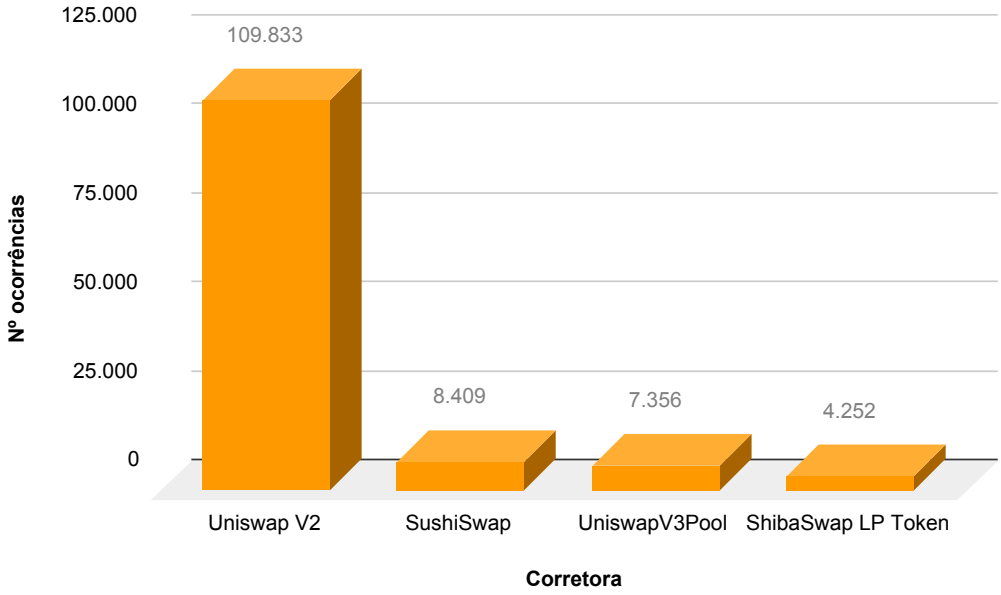

Figura 3. Corretoras com maior quantidade de ocorrências de suspeitas.

Tabela 3. *Bots* mais relacionados com os ataques.

Bot	Qtde.
0x00000000A991C429eE2Ec6df19d40fe0c80088B8	35.465
0xa7003527AF20001c000037A90051B19Ce31EEd36	17.287
0x007933790a4f00000099e9001629d9fE7775B800	15.836
0x00000000500e2fece27a7600435d0C48d64E0C00	14.375

4. Trabalhos Relacionados

Em [Torres et al. 2021], principal referência para este trabalho, os autores apresentam uma metodologia para identificar os três tipos de ataques *front-running*: deslocamento, inserção (também conhecido como ataque sanduíche) e supressão. A partir da análise de mais de 11 milhões de blocos, os autores identificaram quase 200 mil ataques com

um lucro acumulado de 18,41 milhões de dólares para os atacantes durante o período 30 de julho de 2015 até 21 de novembro de 2020. Através dos resultados, os autores demonstram que os ataques *front-running* são lucrativos e um problema predominante na blockchain Ethereum. Em nossa abordagem, expandimos um dos tipos de ataque *front-running*, que é o ataque sanduíche, e a partir de novas suposições relacionadas com as características deste ataque, aplicamos uma análise adaptada para o ano de 2023, com o objetivo de reforçar a importância desta vulnerabilidade para o ecossistema blockchain.

Já em [Varun et al. 2022] é proposto um esquema de detecção e prevenção de ataques *front-running* baseado em modelo. Os autores extraem recursos específicos para cada transação e transformam cada uma em um vetor de recursos que pode ser analisado por um modelo de aprendizado de máquina, a fim de detectar se uma transação é maliciosa ou não em tempo real. Os autores também realizam experimentos em um grande conjunto de dados de transações para verificarem a eficácia da abordagem proposta. Além disso, os autores utilizam o conjunto de dados sobre ataques *front-running* fornecido em [Torres et al. 2021]. Em nosso trabalho, estamos interessados em detectar as transações maliciosas apenas de ataques sanduíche, mas com a finalidade de analisar como e quanto é lucrativo este tipo de ataque, bem como os prejuízos para as vítimas.

Por sua vez, em [Zhang et al. 2023a] é apresentado um algoritmo para a identificação de ataques *front-running*. Os autores demonstram que o algoritmo é eficaz e abrangente na detecção dos ataques em comparação com estudos anteriores. Além disso, os autores propõem uma abordagem de localização de vulnerabilidades para ataques *front-running* em trechos de códigos de contratos inteligentes, de maneira automatizada e escalonável. Os autores ainda avaliam empiricamente sete técnicas de detecção de vulnerabilidades em um benchmark criado por eles. O benchmark é composto pela análise de 800.000 blocos. O experimento conduzido revela a inadequação das técnicas existentes na detecção de vulnerabilidades *front-running*, por conta de limitações como falta de suporte para análise entre contratos, resolução ineficiente de restrições para operações criptográficas, padrões de vulnerabilidade inadequados e falta de suporte de token. Semelhantemente, apresentamos como os ataques sanduíche ainda ocorrem na rede Ethereum e apontamos lacunas em abertos de estratégias de mitigação desta vulnerabilidade.

Existem duas razões principais pelas quais os ataques *front-running* são possíveis em blockchains públicas como o Ethereum, são a falta de confidencialidade das transações e a capacidade dos mineradores ordenarem as transações arbitrariamente. O fato das transações serem transparentes para todos é sem dúvida uma das principais vantagens de uma blockchain pública. No entanto o conteúdo e a finalidade de uma transação só devem ser visíveis para todos depois de ter sido extraída. A ideia de ordenar transações com base no preço do gás é uma boa estratégia à primeira vista, mas isto também introduz determinismo de uma forma que pode ser manipulado por estranhos [Torres et al. 2021].

Em [Bentov et al. 2019] é apresentado o *Tesseract*, uma exchange que é resistente ao *front-running*, aproveitando um ambiente de execução confiável. Os projetos de plataformas centralizadas existentes são vulneráveis ao roubo de fundos, enquanto as plataformas descentralizadas não podem oferecer negociações entre cadeias em tempo real. O *Tesseract* supera essas falhas e alcança um design do melhor dos dois mundos usando um ambiente de execução confiável. No entanto, seu design segue uma abordagem centralizada e exige que os usuários tenham suporte de hardware para uma execução

confiável.

Em [Heimbach and Wattenhofer 2022] os autores generalizam o problema do ataque sanduíche em um *Sandwich Game* para analisar tanto da perspectiva do atacante quanto da vítima. Os autores também apresentam um algoritmo que os comerciantes podem usar para definir a tolerância à derrapagem dos preços. O trabalho mostra que se pode evitar a maioria dos ataques sanduíche com um ajuste da tolerância à derrapagem.

5. Considerações Finais

Os ataques sanduíche exploram a fila de transações pendentes de redes blockchain públicas, i.e., *front-running*, especificamente transações de grande valor financeiro, cercando essas transações com outras de origem maliciosa. Neste trabalho, analisamos potenciais ataques sanduíches durante o ano de 2023, por meio da coleta de transações efetivadas na rede Blockchain Ethereum. A partir dos dados coletados, analisamos definições já existentes na literatura sobre o ataque sanduíche, assim como estendemos essas definições, com o objetivo de demonstrar como esse tipo de ataque ocorre recentemente na rede Ethereum.

Nossos resultados evidenciam a necessidade de uma definição mais ampla sob ataques sanduíches para incluir as tentativas de ataques cujas estratégias de tarifações não são claras, e ainda assim ocasionam lucro para os atacantes e potenciais perdas para os demais usuários (i.e., vítimas), o que vem a reduzir a confiança no ecossistema DeFi. Nesse sentido, propomos as perspectivas abrangentes e restritas sob ataques sanduíches. A primeira considera um perfil de atacante mais propenso a riscos, independente de custos de tarifações para especular ganhos com o token alvo. Sob essa perspectiva nossas medições apontam um lucro médio de USD 3.202,82, mas uma tendência de prejuízos para cerca de 50% dos atacantes. Por sua vez, na perspectiva restrita, i.e., atacantes que manipulam a *mempool* via o mecanismo de tarifação conforme descrito na literatura, observamos lucro médio cai pela metade (USD 1801.96) mas menos de 25% dos ataques têm prejuízos. Ambos os casos são danosos ao ecossistema DeFi, mas ataques sob a perspectiva abrangente são ainda piores, dado que observamos sua tendência de crescimento em 2023, em comparação à perspectiva restrita que tende à queda.

Como trabalhos futuros, planejamos realizar uma análise mais profunda sobre as transações, com o objetivo de explicar detalhadamente a variação do preço do gás entre as transações e eliminar falsos positivos. Por meio dos resultados da perspectiva abrangente, existem ataques que foram realizados sem a necessidade de pagar um preço maior de gás na primeira transação do atacante, podendo ser indícios de possíveis conluios com mineradores da rede. Ademais, pretendemos propor estratégias de mitigação deste tipo de ataque levando em consideração o ecossistema DeFi, a arquitetura da blockchain e contratos inteligentes, bem como as técnicas de segurança difundidas na literatura.

6. Agradecimentos

Este trabalho foi realizado com apoio financeiro do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) processo no. 88887.918434/2023-00 e Fundação de Amparo à Pesquisa do Piauí (FAPEPI) processo no. 00110.000235/2022-78.

Referências

Bentov, I., Ji, Y., Zhang, F., Breidenbach, L., Daian, P., and Juels, A. (2019). Tesseract: Real-time cryptocurrency exchange using trusted hardware. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 1521–1538, New York, NY, USA. Association for Computing Machinery.

Chen, T., Li, Z., Zhu, Y., Chen, J., Luo, X., Lui, J. C.-S., Lin, X., and Zhang, X. (2020). Understanding ethereum via graph analysis. *ACM Trans. on Internet Technology (TOIT)*, 20(2):1–32.

Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., Breidenbach, L., and Juels, A. (2019). Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges.

Eskandari, S., Moosavi, S., and Clark, J. (2020). Sok: Transparent dishonesty: Frontrunning attacks on blockchain. In Bracciali, A., Clark, J., Pintore, F., Rønne, P. B., and Sala, M., editors, *Financial Cryptography and Data Security*, pages 170–189, Cham. Springer International Publishing.

Harvey, C. R., Ramachandran, A., and Santoro, J. (2021). *DeFi and the Future of Finance*. John Wiley & Sons.

Heimbach, L. and Wattenhofer, R. (2022). Eliminating sandwich attacks with the help of game theory. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '22. ACM.

Qin, K., Zhou, L., and Gervais, A. (2021). Quantifying blockchain extractable value: How dark is the forest? *CoRR*, abs/2101.05511.

Torres, C. F., Camino, R., and State, R. (2021). Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1343–1359. USENIX Association.

Varun, M., Palanisamy, B., and Sural, S. (2022). Mitigating frontrunning attacks in ethereum. In *Proceedings of the Fourth ACM International Symposium on Blockchain and Secure Critical Infrastructure*, BSCI '22, page 115–124, New York, NY, USA. Association for Computing Machinery.

Wang, Y., Zuest, P., Yao, Y., Lu, Z., and Wattenhofer, R. (2022). Impact and user perception of sandwich attacks in the defi ecosystem. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Weintraub, B., Torres, C. F., Nita-Rotaru, C., and State, R. (2022). A flash(bot) in the pan: measuring maximal extractable value in private pools. In *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, page 458–471, New York, NY, USA. Association for Computing Machinery.

Xu, X., Weber, I., and Staples, M. (2019). *Architecture for blockchain applications*. Springer.

Zhang, W., Wei, L., Cheung, S.-C., Liu, Y., Li, S., Liu, L., and Lyu, M. R. (2023a). Combatting front-running in smart contracts: Attack mining, benchmark construction

and vulnerability detector evaluation. *IEEE Transactions on Software Engineering*, 49(6):3630–3646.

Zhang, Y., Liu, P., Wang, G., Li, P., Gu, W., Chen, H., Liu, X., and Zhu, J. (2023b). Frad: Front-running attacks detection on ethereum using ternary classification model. *arXiv preprint arXiv:2311.14514*.