文章编号:1671-4229(2022)02-0001-15

DeFi安全:攻击、检测与防御初探

罗瑞杰, 王炳森, 宋书玮, 姜人楷, 罗 丰,
林煌坤, 张小松, 陈 厅*

(电子科技大学 网络安全技术实验室, 四川 成都 611731)

摘 要: 自区块链技术提出以来,以太坊作为区块链2.0的代表,它的高速发展促进了去中心化金融DeFi的繁荣。但是在DeFi生态进一步发展的同时,也遭遇了许多的安全攻击,造成了大量的损失,因此DeFi安全是值得研究的热点问题。然而DeFi攻击的方式多变、原因繁杂,现有的攻击分类方法难以清楚地阐明攻击的原理和发生的原因。文章基于DeFi通用架构模型,提出了一种新的分类方法,将攻击归类到不同的DeFi层次,通过对各类攻击流程的详细研究,分析造成各个攻击的具体原因,以及对应的DeFi层次的安全风险和漏洞。文章进一步总结了现有的检测和防御方法,结合各个方法的优缺点和DeFi本身的特点对现有方法进行改进。对于有明显缺陷的检测防御方法,提出新的解决方案,并指出了未来DeFi发展下安全问题的研究方向。

关键词: 区块链; 去中心化金融; 安全; 漏洞

中图分类号: TP 391 **文献标志码**: A

DeFi security: A preliminary exploration of attacks, detection and defense

LUO Rui-jie, *WANG Bing-sen*, *SONG Shu-wei*, *JIANG Ren-kai*, *LUO Feng*,
LIN Huang-kun, *ZHANG Xiao-song*, *CHEN Ting**

(Center for Cybersecurity, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract: Since the introduction of blockchain technology, Ethereum, as the representative of blockchain 2.0, its rapid development has promoted the prosperity of decentralized finance DeFi. However, while the DeFi ecosystem is further developing, DeFi has encountered many security attacks in recent years, causing a lot of losses due to defects in some related technologies. Therefore, DeFi security is a hot issue worth studying. DeFi security can be classified into blockchain security, which is a relatively new field in blockchain security. Many DeFi attacks have evolved from common blockchain attacks. However, the methods of DeFi attacks are changeable, complex, and involve a wide range of areas. It is difficult for the existing attack classification methods to clearly explain the attack principle and the related vulnerabilities involved in the attack. This paper is based on a general DeFi architecture model, which is divided into settlement layer, asset layer, protocol layer, application layer and aggregation layer from bottom to top. Adjacent layers interact with each other to jointly complete the related services of decentralized finance. There are many types of DeFi applications with different forms. Using this general model can better explain the basic principles and architecture of DeFi. This

基金项目: 国家重点研发资助项目(2018YFB0804100); 国家自然科学基金资助项目(61872057, U19A2066); 四川省科技资助项目(2020JDTD0007)

作者简介: 罗瑞杰(1998—),男,硕士研究生. E-mail: jerryluorj@outlook.com

* 通信作者. E-mail: brokendragon@uestc.edu.cn.

引文格式: 罗瑞杰, 王炳森, 宋书玮, 等. DeFi安全:攻击、检测与防御初探[J]. 广州大学学报(自然科学版),2022,21(2):1-15.

paper proposes a new classification method to classify attacks into different DeFi levels. Through a detailed study of various attack processes, it analyzes the specific reasons for each attack, as well as the security risks and vulnerabilities of the level. This paper further summarizes the existing DeFi attack detection and defense methods. Most of the detection methods use traditional blockchain security methods, which are difficult to adapt to the high openness and composability of DeFi applications. This paper combines the advantages and disadvantages of each method and the characteristics of DeFi itself to improve the existing methods. For detection and defense methods with obvious flaws, new solutions are proposed. At the same time, it also pointed out the future development of DeFi and the corresponding research direction of security issues.

Key words: blockchain; DeFi; security; vulnerability

近年来,去中心化金融 DeFi(Decentralized Finance)应用在各类公共区块链上发展迅猛,特别是以太坊这类,具有运行预定义自动化程序(通常称为智能合约)的公链,给 DeFi 的发展提供了广阔的前景[1]。去中心化应用利用区块链的透明性和开放性,在公链上运行具有各类具有金融功能的智能合约,提供了包括代币交换、数字资产抵押等多样化的金融服务[2]。DeFi 应用相比传统金融服务提供商在效率、透明性、可访问性以及可组合性上有着显著的优势[3]。各类更加高效、更具有创新性的去中心化金融协议的提出,让 DeFi 应用进入高速发展的时期。截至 2022 年 2 月,锁定在 DeFi 应用上的总资产达到了 1 500 亿美元,而在 2021 年 DeFi 的资产增长率更是达到了惊人的 1 700%[4]。

DeFi 的快速增长,在带来了巨大收益并使相关区块链技术进一步发展的同时,也面临着严峻的安全挑战。由于 DeFi 应用中存在着不少的漏洞,给黑客提供了各类窃取 DeFi 应用资产的手段。以太坊是 DeFi 应用的主要运行环境,据 TokenInsight 的行业报告显示[5],截至 2021 年 12 月,DeFi 应用中 63% 属于以太坊,现有攻击大多数是针对以太坊的 DeFi 应用。本文主要对以太坊上的攻击情况进行分析,同时有 PolyChain,币安链等新兴公链的攻击研究。在过去的 2021 年里,DeFi 上资产的迅速上涨,极大地提高了安全威胁带来的风险。在 2021 年 8 月,DeFi 平台 PolyNetWork 被盗超过 6 亿美元的加密资产[6],而在 2022 年 2 月,Wormhole 平台被盗取超过 3 亿美元的加密资产[7]。截至 2021 年 12 月,在 DeFi 上共有超过 120 亿美元的资产遭受损失[8]。其中,仅 2021 年,被盗取和欺诈的 DeFi 资产达到了 105 亿美元,远远高于 2020 年的 15 亿美元[8]。这些攻击都给 DeFi 平台以及相关用户带来了巨大的损失,严重影响了 DeFi 系统的生态环境。

学术界中关于 DeFi 的安全研究较少且主要是针对某一具体类型的攻击进行分析。Qin 等[9]详细阐述了 DeFi 应用中出现的套利攻击,对其进行了定量分析,并量化了套利行为可能出现的风险。Wang 等[10]提出了一种可扩展的安全分析框架,可以检测 DeFi 应用中存在的价格预言机漏洞。Tolmach 等[11]提出一种形式化的过程代数技术,分析 DeFi 协议组合后存在的经济以及安全方面的问题。Wang 等[2]在智能合约漏洞检测的基础上,提出了一种挖掘漏洞的工具,可以对 DeFi 的智能合约进行相关分析,并通过监控 DeFi 所在公链的相关交易来进行攻击检测。学术界关于 DeFi 的安全研究大多是在已有智能合约漏洞的基础上进行的[2],或是针对某一类具体的攻击进行检测防御[9-11]。由于 DeFi 攻击方式多变,现有关于 DeFi 的安全检测和防御方法使用的传统区块链安全中的方法,只能针对某一具体类别的攻击。本文总结了现有关于 DeFi 的各类安全检测和防御方法,使用层次化的方式研究 DeFi 安全问题。通过将不同的攻击分类到所属的 DeFi 架构层次中,依据 DeFi 层次的特点对现有检测和防御方法进行划分,同时提出新的方法用于改进。区别于原有的 DeFi 安全研究,本文通过层次化的方法,可以更加全面地解决现有的各类安全攻击问题,并对 DeFi 系统中存在的安全漏洞提出了解决或改善方法,有效地防止了各类可能存在的 DeFi 安全攻击。

DeFi 自应用以来,国内外已有多篇关于 DeFi 的综述。如 Jensen 等[12]研究了 DeFi 市场分类和常见用例,并指出了潜在的一些关键风险。Schär[3]对 DeFi 的架构以及常见类型进行了一个较为完整的总结,并针对可能存在的安全漏洞做出了相关分析。Katona[13]的研究对 DeFi 进行了更加准确和全面的定义,并对 DeFi 的相关收益进行了分析。Werner 等[14]则从协议层面对 DeFi 进行分类并分析协议可能带来的技术和经济风险。Kim 等[15]分析了现有 DeFi 的常见应用,并对常见类型进行了分组。Andrea 等[16]分析了常见的 DeFi 应用,并抽象出了各个类型应用中的关键模型,提出了这些模型涉及

到的技术以及非技术的安全问题。Corbet 等[17]研究了 DeFi 代币和传统加密货币(如比特币等)的区别,并研究了 DeFi 上出现的市场波动等情况。Popesce 等[18]对比了和传统金融的关系后,总结了 DeFi 应用新增加的各类金融功能。Qin 等[19]除了对比与传统金融的关系外,着重研究了 DeFi 可能出现的经济安全风险。由于 DeFi 发展时间较短,现有综述主要是对 DeFi 应用的种类进行研究,关于 DeFi 安全的讨论则层次较为单一,通常是按照不同 DeFi 的协议或功能分析背后的安全问题[14,16]。本文与现有的 DeFi 安全综述不同,总结了大量的真实攻击案例,并首次按照 DeFi 的架构模型进行分层分类,系统探究了 DeFi 生态系统中存在的各类安全风险。本文在分类的同时,首次对每类攻击的典型案例进行了详细分析,并对攻击流程中涉及到的安全漏洞进行了分析,从 DeFi 架构的各个层次对安全问题进行较为全面的探究,同时本文总结了已有 DeFi 攻击的检测和防御方法,并在此基础上进行了改进。

本文首先总结了自 DeFi 提出以来的各种攻击案例,依据 DeFi 的层次架构,对不同的 DeFi 攻击进行了分类总结。随后本文探究了各类攻击对 DeFi 生态系统造成的影响,选取了各类攻击中的典型案列进行详细的流程说明,以此挖掘攻击背后涉及到的安全漏洞。在完整的对所有攻击分类后,本文总结了现有针对 DeFi 攻击的各种检测和防御手段,分析各个方法的优缺点,之后结合 DeFi 各层次的特点对现有方法进行改进,或提出新的解决方案。最后,本文探索了未来 DeFi 安全中检测和防御方法的研究重点,并对全文进行了总结。本文主要有以下 3 点贡献:①总结了自 2020 年来至今(2022 年 2 月)的所有真实攻击案例,并首次按照层次对 DeFi 攻击进行了系统的分类;②首次对每一类攻击的典型案例进行了详细的流程分析,并挖掘了 DeFi 架构下存在的各类安全漏洞和风险;③总结了 DeFi 安全的各类检测和防御方法,分析现有方法的优缺点,并对方法进行改进或提出新的解决方案。

本文的组织安排:第 1 节介绍了区块链相关概念,DeFi 应用的通用模型及层次划分,以及各层次对应的安全风险;第 2 节针对 DeFi 攻击进行分类,并提出分类的依据和方法;第 3 节针对攻击的分类,总结已有检测和防御方法的优缺点,提出改进措施或者提供新的解决方案;第 4 节分析 DeFi 安全问题的发展趋势以及未来研究方向;第 5 节总结全文。

1　DeFi 架构及安全问题

本节介绍区块链和 DeFi 的相关概念,以及 DeFi 的通用模型,分析该模式下的各类安全风险。

1.1　区块链

区块链是一种分布式数据库解决方案,它维护着一个不断增长的数据列表,并由参与其中的节点确认记录[20]。区块链的相关概念由中本聪在 2008 年首次提出[21],并在近年来不断发展完善,现有的比特币和以太坊是区块链技术的主要代表。区块链技术涉及多个领域的知识,其数据的所有权和隐私安全依赖于密码学来保障,而随着区块链不断发展,以及区块链相关领域应用价值的不断增大,区块链相关安全的研究变得更加重要。

1.2　智能合约

智能合约是一种运行在以太坊链上的程序。可以由高级语言开发并编译成以太坊虚拟机字节码,智能合约在部署后不能修改。在执行智能合约时,虚拟机会维护一个堆栈、内存和存储数据的账户存储[22]。为了防止滥用资源,智能合约的部署和调用将向交易发送者收取费用。

1.3　汽油

汽油(gas)是用来衡量一笔交易所消耗的计算资源的基本单位[23]。当用户进行交易时,以太坊虚拟机会计算这笔交易中所使用的资源,交易的发起者需要支付执行交易的费用。

1.4　DeFi 定义

去中心化金融 DeFi 可以定义为建立于公共智能合约平台上的开放无需许可且高度可互操作的的协议栈,它以更加开放和透明的方式实现传统的金融服务[3]。DeFi 应用是传统金融依托区块链的去中心化实现,DeFi 与传统金融存在一些相似之处,DeFi 应用的类别也来源于传统的金融服务类别。现有的 DeFi 应用中,主要提供有借贷、交换、抵押、存储、投资、保险等金融业务。DeFi 应用依托于支持智能合约运行的区块链,主要是以太坊为主,其他如 EOS, TRON, 以及 Cosmos 等链的 DeFi 应用也在逐步发展中[13]。

1.5　DeFi 应用

DeFi 应用的类型较多,其中主要包括了数字资产交换,以及代币抵押。DeFi 的数字资产交换主要通过自动做市商(automated market maker, AMM)实现。AMM 是一类 DeFi 应用,它将一或多对的数字资产记录后作为流动性代币池(实质是智能合约),代币池中存有相应的数字资产[24]。买卖双方可以直接与流动性代币池进行交互,代币池使用数学公式计算流动池中的代币价格[25],并使用经济模型维持代币池的交易市场。AMM 改变了传统买卖双方订单匹配的模式,使用去中心化的

方法完成交易。

1.6 DeFi 通用架构模型

DeFi 的生态系统,以一种高度自由和透明的方式构建。DeFi 中不同应用提出的协议以及程序本身都可以组合以提供更好的金融服务。Schär[3] 提出了一种通用概念模型,该模型通过分层的方式很好的解释了 DeFi 协议的运行及交互的方式,突出了 DeFi 系统中关键的组合特性。下文详细介绍该模型下的 DeFi 分层,并针对各层特点提出面临的风险以及安全漏洞。DeFi 的架构模型,如图 1 所示。

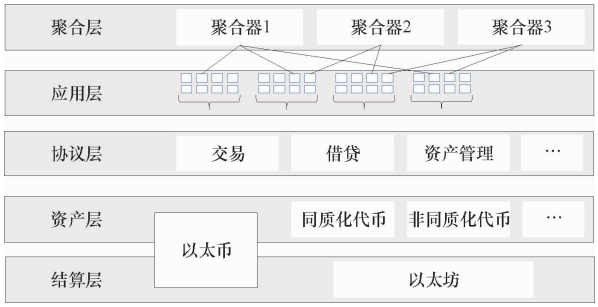

图 1 DeFi 通用架构模型图

Fig. 1 DeFi general architecture model diagram

(1)结算层。结算层是 DeFi 应用最底层的实现。由于 DeFi 应用需要依托于智能合约来实现其相应的金融服务功能,因此,结算层通常是允许智能合约运行的区块链组成,而 DeFi 系统的最终交易也是在结算层实现。结算层除了存在传统区块链的安全风险,如双花攻击、日蚀攻击等,针对于 DeFi 特性而言,结算层更为重要的漏洞在于区块链的交易顺序。DeFi 中的各类金融活动中,许多都依赖具体的交易顺序,不同金融活动在不同的交易顺序下,会有很大的差异。而区块链中的矿工或者节点可以通过条件竞争的方式[26],改变具体的交易,使用套利的行为来实现获利[9]。

(2)资产层。资产层在结算层之上,由结算层发行的加密货币构成,如 ETH、BNB 等货币。该层的资产包括依托于区块链的原生代币,以及其他依托于智能合约的代币。由于代币中存在各类标准,但是仍然存在许多有缺陷的代币[27],这和 DeFi 应用的智能合约存在接口不一致,导致出现漏洞,造成 DeFi 平台资产的损失。

(3)协议层。协议层通常由实现了各类金融服务的智能合约构成,DeFi 应用各种功能的运行逻辑在协议层实现。协议层是 DeFi 模型中最为主要的层次,其中的协议通常运行于智能合约,由各个 DeFi 应用的开发者提出运行于区块链中。由于协议主要由智能合约构成,因此,过去的智能合约漏洞也会对 DeFi 的生态系统造成威胁,比如常见的重入漏洞、权限验证漏洞、变量覆盖漏洞等都会影响到 DeFi[28],且现有的大量 DeFi 攻击都是针对协议层存在的漏洞进行攻击。

(4)应用层。DeFi 应用的应用层是用户和 DeFi 平台交互的接口,通常依托于传统 Web 浏览器前端,以网页等方式呈现。DeFi 应用与传统 Web 应用类比,也可以分为前后端。DeFi 应用的后端建立区块链上,但前端需要使用传统的 Web 应用的构建方法,这就给 DeFi 应用带来了较大的风险。DeFi 应用前端受传统网络攻击手段的影响,存在有诸如 XSS、CSRF 漏洞攻击、网络劫持等漏洞[29],前端受到攻击不会直接对 DeFi 的协议层造成影响,但是造成的漏洞被攻击者利用会造成 DeFi 应用或用户资产的损失。

(5)聚合层。聚合层用于实现 DeFi 生态系统的可扩展性,通过聚合层不同开发者实现的 DeFi 应用得以组合交互。聚合层主要将不同的 DeFi 协议关联连接,使其得以交互,在实现更复杂功能的同时,呈现出相对简洁的形式。由于不同的协议有不同的实现标准,在通过聚合层进行交互时可能会出现协议之间不兼容的情况,造成使用金融服务时出现漏洞,攻击者利用漏洞造成 DeFi 生态系统的破坏。

2 DeFi 攻击分类

2.1 结算层中的攻击分类

预言机攻击:预言机是一种单向的数字代理,可以查找和验证真实世界的数据,并以加密的方式将信息提交给智能合约,相当于区块链世界中的一个第三方数据代理商[30]。预言机是区块链网络与互联网以及其他区块链网络等保持数据、信息沟通的桥梁。特别是在 DeFi 智能合约这类去中心化应用(Dapp)中,通过预言机,开发者可以调用包括行情价格在内的各种外部数据资源,让 Dapp 连通外部现实世界的数据环境。

2021 年以来,以太坊 DeFi 生态中有 10 多起利用闪电贷的大规模攻击事件陆续被媒体曝光,而这些事件已经呈现出明显的模式化和重复性特征,从表面上看,其共性是有些 AMM 协议会作为 DeFi 智能合约的预言机报价源,而攻击者通过操纵 AMM 资产池内的资产价格或者资产数量来造成损失。攻击流程是依赖预言机报价的系统,会临时操纵报价以扭曲协议的内部核算,然后将资金以优惠的利率存入,再将预言机重置为正常值后立即以另一种货币或同一种货币提走。通常黑客们选择使用闪电贷(即允许用户零抵押贷出巨额资产,但必须在同一个区块内还款,否则交易会回滚)来获取攻击所需的巨额筹码。黑客通过一系列手段出入各类抵

押、借贷、交易协议,利用巨额资金扭曲某个单一市场的价格数据,进而扰乱预言机报价结果,最终实施套利。攻击原理[31],如图 2 所示。

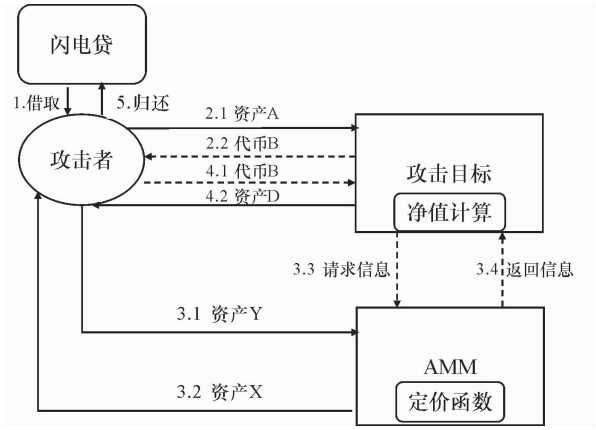

图 2　预言机攻击流程图

Fig. 2　Oracle attack flow chart

(1)准备,获得用于操纵 AMM 预言机的资产 Y 及准备存入攻击目标的资产 A;

(2)抵押,将资产 A 抵押至攻击目标,获得代表抵押物的头寸代币 B,有些情况下,不发放头寸代币 B,只在智能合约内部记账;

(3)操纵,将资产 Y 投入 AMM 兑换资产 X,改变 AMM 流动性池内资产的比例从而改变报价,更新攻击目标合约内资产 A 或头寸代币 B 的定价;

(4)收尾,若为借贷业务,则通过抬高的抵押物估值借出更多资产并不再归还;若为机枪池业务,则通过抬高价格后的头寸代币 B 赎回资产,获得增值收益。

2020 年 12 月 18 日,DeFi 借贷协议 WarpFinance 遭到黑客攻击[32],造成了近 800 万美元的资产损失。其攻击过程如下:

(1)Warp Finance 使用的是 Uniswap 交易对的相对价格作为其预言机的报价源;

(2)攻击者在了解到这个情况后,使用从闪电贷中获取的巨额资金操纵了 Uniswap 交易对的价格;

(3)通过控制预言机报价源信息,攻击者破坏了 Warp Finance 的借款价值判断标准;

(4)在 Warp Finance 错误的数据环境下,攻击者窃取了远远超过抵押品价值的资产;

(5)攻击者归还了从闪电贷中借出的款项。

三明治攻击:三明治攻击是 DeFi 里流行的抢先交易技术的一种。为了形成一个“三明治”交易,攻击者会找到一个待处理的受害者交易,然后试图通过前后的交易夹击该受害者。这种策略来源于买卖资产从而操作资产价格的方法[33]。区块链的透明度以及执行订单的延迟(往往在网络拥堵情况下),使抢先交易更加容易,并极大降低了交易的安全性。

所有区块链交易都可在内存池中查到。一旦掠夺性交易者注意到潜在受害者的待定资产 X 交易被用于资产 Y,他们就会在受害者之前购买资产 Y。掠夺性交易者知道受害者的交易将提高资产的价格,从而计划以较低的价格购买 Y 资产,让受害者以较高的价格购买,最后再以较高的价格出售资产。

实现三明治攻击的 3 个要素点[34]:①交易公开性,可以在内存池中获取交易的详细信息;②以太坊交易执行机制,通过汽油竞争的方式抢先完成交易,用户可以提高汽油价格来优先让矿工进行打包交易;③AMM 交易曲线机制,通过恒定乘积机制可以造成较大滑点来使用户受损。

2.2　资产层中的攻击分类

平台与代币不一致攻击:代币资产通过 DeFi 进行交易,但 DeFi 和代币之间的交互方式却不一定能保证正确。不正确的交互可能导致不一致的行为。现有交易中的大多数情况都默认 DeFi 和代币的交互是一致的,这就可能会导致不一致攻击的发生。例如攻击者利用某种有缺陷的代币 A 作为资产与 DeFi 交互,然而,DeFi 平台默认代币 A 是正常资产,因此,在成功调用转账接口后,错误的对资产进行了记录,同时并未检查实际资产情况,导致出现不一致,造成财产损失和用户混乱。

不一致的定义:用户用代币在 DeFi 平台进行交互时,代币的行为与 DeFi 平台的行为不匹配。M 代表存储用户余额在 Token 合约中的核心数据结构,N 代表存储每个用户余额记录在 DeFi 合约中的核心数据结构,将每个用户的余额记录存储在一个 DeFi 合约。为了证实表征不一致,让 Bm 表示修改 M 的行为。假设 Bn 表示修改 N 的行为,通过比对 Bm、Bn 来判断是否不一致。

2020 年 6 月 28 日,DeFi 应用 Balancer 遭受攻击[35]。Balancer 是一个提供 AMM 服务的合约,也就是自动化做市商服务,自动化做市商服务提供者采用代币池中的各种代币之间的数量比例确定代币之间的价格,用户可通过这种代币之间的动态比例获取代币之间的价格,进而在合约中进行代币之间的兑换。攻击原理如图 3 所示。

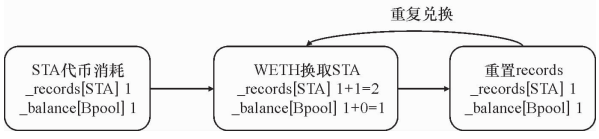

图 3　资产不一致示意图

Fig. 3　Schematic of asset inconsistency problem

攻击过程如下:

(1)从 借贷平台 dYdX 进行贷款;

(2)不断地调用 swapExactAmountIn()函数,将 Balancer 池中的 STA 数量降到低点,推高 STA 兑换 其他代币的价格;

(3)使用1个STA 兑换 WETH,并在每次兑换完成后(调用 swapExactAmountIn()函数)调用 gulp()函数,覆盖 STA 的余额,使 STA 兑换 WETH 的价格保持在高点;

(4)使用同样的方法攻击代币池中的其他代币;

(5)偿还闪电贷贷款;

(6)获利离场。

此次攻击利用了 STA 代币的特殊性从而导致的不一致行为,STA 代币是通缩型代币,使用 1 个 STA 进行兑换,转账过程中着 1 个 STA 被燃烧掉,实际上 Balancer 收到 0STA,将交易池中 STA 的数量始终维持在一个很低的水平。产生不一致攻击的主要原因在于 STA 合约记录的代币更新量与 balancer 合约记录的代币更新量不一致。当用户在使用通缩型代币进行兑换的时候,Balancer 合约没有有效地对接收到的通缩型代币的实际余额进行校验,导致余额记录错误。

2.3 协议层中的攻击分类

重入攻击:攻击者利用 DeFi 智能合约存在的重入漏洞,通过自定义的恶意合约多次重复调用平台关键操作,造成平台和用户的资产损失。智能合约中的重入漏洞官方定义为:从智能合约 A 与任意不同的智能合约 B 的交互,以及 A 与 B 之间发起的以太币转移,都会将控制权移交到 B,同时 B 拥有在交互结束前回调合约 A 函数的能力[36]。重入漏洞是智能合约中的典型安全漏洞,由于 DeFi 应用中会涉及到多个合约,且具有较多的外部调用和交互[37],因此,DeFi 项目中重入攻击较为严重。

已有重入攻击中,有直接针对 DeFi 平台资产记录的重入攻击,针对平台关键计算变量的重入攻击,以及针对平台代币转账的重入攻击。重入攻击在 DeFi 项目中造成了较为严重的损失,Uniswap V1[38]、Akropolis[39]、BurgerSwap[40]等大型 DeFi 平台均遭受了重入攻击。截至 2021 年 12 月,已报道的 DeFi 重入攻击至少造成了约 2 400 万美元的损失。

针对 DeFi 项目的重入攻击要造成经济损失,需要满足至少 2 个条件:首先是 DeFi 关键函数存在重入漏洞。这里的关键函数主要指涉及平台资产计算、存储、更新操作的函数。这类函数如果存在重入漏洞,攻击者可以通过多次调用关键函数,造成平台资产的非法变动,导致经济损失;其次是 DeFi 应用关于外部合约的限制情况。如果平台不限制代币合约类型以及地址,导致攻击者可以使用恶意合约伪装为资产传入平台,并在 DeFi 应用上使用含有重入漏洞的函数进行外部调用时,发起重入攻击。

2021 年 5 月 18 日, DeFi 项目 BurgerSwap 遭受攻击,损失达 330 万美元[40]。该项目属于 DeFi 应用中的自动做市商 AMM 类型,项目架构主要由 Delegate 层,lpPlatForm 层和 Pair 层构成,项目结构如图 4 所示。

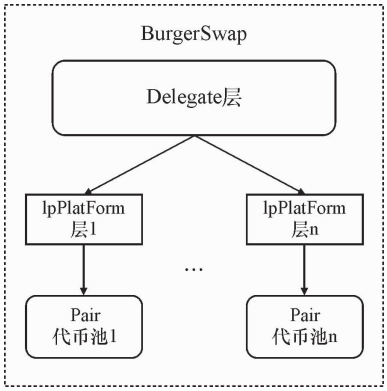

图 4 BugerSwap 架构示意图

Fig. 4 Schematic of BugerSwap architecture

用户在 BurgerSwap 交易时,需要确定 2 种资产代币,由 Delegate 层判断 2 种代币是否已经建立代币池合约。如果还未建立,由 Delegate 层产生对应资产对的 lpPlatForm 层和 Pair 层,其中,lpPlatForm 层进行逻辑运算,同时掌握 Pair 层的权限,可以发起转账并记录用户资产情况,Pair 层合约则存储实际资产。此架构下的 IpPlatForm 层中,由于实际转账前未核算 Pair 层资产实际情况,完全按照函数参数和平台记录进行逻辑运算,导致出现了重入漏洞,构成了威胁。攻击者利用该漏洞,发起了攻击,攻击过程如图 5 所示。

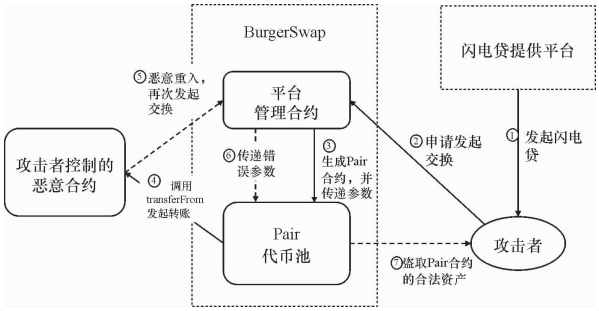

图 5 重入攻击流程图

Fig. 5 Reentrancy attack flowchart

(1)攻击者申请闪电贷,获取 WBNB 代币作为攻击资本;

(2)攻击者发起资产交换请求,要求用自己的恶意代币换取 WBNB 代币;

(3)BurgerSwap 平台建立恶意合约代币与 WBNB 流动性池子,并调用函数计算相关转账参数开始交换;

(4)平台开始发起实际转账,调用了恶意合约的 transferfrom( )函数;

(5)恶意合约发起重入操作,再次向 BugerSwap 发起资产交换请求;

(6)回到原函数,最开始的交换操作已经改变流动性池子资产,资产兑换比例理应发生变化,而后续的交换操作使用原有比例而检查实际比例,导致用较高比例兑换出 WBNB;

(7)攻击者通过重入以高比例兑换出 BugerSwap 的 WBNB 代币,实现从 DeFi 平台盗取资产。

参数校验攻击:参数检验攻击主要是分为函数权限设置问题和未正确校验传入参数 2 种[41],在 DeFi 与 Token 交互中双方都可能会出现的问题,函数权限设置问题通常是由于合约开发者的疏忽导致,很多内部函数在运行时会直接更改合约存储数据,而不进行相关的检测,如果这部分函数的可见性被设置为 public 或者 external,将产生重大的漏洞。未正确校验传入参数问题可能导致函数按照不可预想的结果执行,从而导致攻击者可以通过事先设计好的交易顺序来获利。比如 permit( )函数如果未做零地址校验,且对应代币的销毁代币方式是将代币发送至零地址,那么攻击者可以转移零地址中被销毁的代币。还例如在一些智能合约中会存在 freeze( )函数,用于冻结账户,但是在进行代币转账时,只验证了来源账户,未对转入地址进行验证导致转入的代币无法提出,还需注意的有 transferFrom( )要额外验证 from 地址。黑名单验证也有类似问题。

截至 2022 年 3 月,参数校验攻击 DeFi 中造成了大量的损失,MonoX Finance[42]、DeFi Saver[43]、Superfluid[44]等平台均遭受攻击,至少造成了 4 400 万美元的损失。

2021 年 11 月 30 日,DeFi 平台 MonoX Finance 遭遇攻击,本次攻击中约合 1 820 万美元的 WETH 和 1 050 万美元的 MATIC 被盗,其他被盗 Token 包括 WBTC、LINK、GHST、DUCK、MIM 和 IMX,损失共计约 3 100 万美元[42]。本次攻击造成的漏洞主要有 2 个,第一个是利用了 swap 合约里没有对池中传入和传出代币作检查,从而利用价格更新机制的问题;第二个是移除流动性的函数未对调用者进行检测,任何用户都可以移除提供者的流动性,使得攻击者传入和传出代币相同时,价格被二次计算并覆盖,导致代币价格不断被推高,并以此代币换出池中的其他代币来获利。攻击过程如图 6 所示。

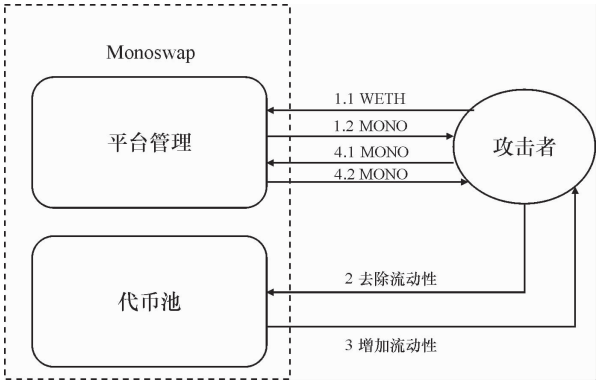

图 6　参数校验攻击流程图

Fig. 6　Parameter verification attack flow chart

(1)黑客将 0.1WETH 兑换为 79.9MONO 作为启动资金;

(2)黑客移除了 pool 内全部流动性,防止攻击受到影响或者价格波动被检测到,此处没有用户的权限进行校验,使得任何人都可以去除流动性;

(3)黑客添加了自己控制的流动性,便于兑换操作;

(4)黑客通过重复 MONO 兑换 MONO 的操作,不断抬高 MONO 价格,此处没有进行对换入换出的代币地址进行校验,可以同种代币进行交换;

(5)利用已经被抬高的 MONO 兑换 pool 内其他资产达到获利目的。

2.4　应用层中的攻击分类

权限攻击:DeFi 权限攻击来源于用户向平台批准 ERC20 代币的无限授权,无限授权会导致攻击者有机会可以通过存在有漏洞的 DeFi 合约盗取用户所有该类 ERC20 代币,而不仅是盗取用户存在 DeFi 的代币资产。大多数 DeFi 应用中,都会默认要求用户赋予所存入代币的无限权限。原因在于,DeFi 应用集合了各类资产代币,其中最为广泛的是使用以太坊 ERC20 标准的代币。而在 ERC20 代币中,如果用户要进行资产转出前,都会检查调用者的权限大小和交易资产数额是否匹配,即需要用户批准相应转账的权限[45]。通过无限权限批准,可以让用户的 DeFi 交易变得更简便,同时也降低了每次赋权所需的交易费用。不过无限授权的形式却会带来极大的风险。

已有利用无限权限的攻击中,大多数与其他攻击结合,呈现一种复合攻击的形式。权限攻击可以和常见漏洞攻击结合,攻击者如果能利用平台漏洞进行代币盗取

时,且恰好具有无限授权,则可以将用户的该类资产全部盗取。同时,在一些遭受钓鱼攻击的DeFi以及一些恶意DeFi项目中,攻击者都会通过无限授权的方式来盗取用户更多的代币,造成更大的损失。权限攻击扩大了DeFi项目遭受的损失,截至2021年12月,Furucomb[46]平台被黑客利用漏洞的同时,使用权限攻击造成了巨量损失,UniCat等恶意DeFi平台则使用权限攻击增加骗取的资产[47],权限攻击至少造成了约4 000万美元的损失。

权限攻击的关键在于用户向DeFi平台的无限授权,而这一现象在DeFi应用中是普遍默认的。同时,权限攻击由于其复合形式的特性,权限攻击还需要DeFi本身具有可攻击的漏洞。如果DeFi本身没有安全问题,则无限授权的方式是有利于用户的使用体验的,因此,更多的DeFi项目正在寻找便利和安全两者的一个平衡点。

2021年2月28日,DeFi项目Furucombo发生了权限攻击,造成超过1 400万美元的损失[46]。该平台主要使用代理合约来完成各类资产的交换,用户存入资产后,使用平台建立的代理合约进行各类金融活动。由于该平台要求用户进行无限授权,而同时平台出现了一个验证漏洞,最终导致了攻击,攻击过程如图7所示。

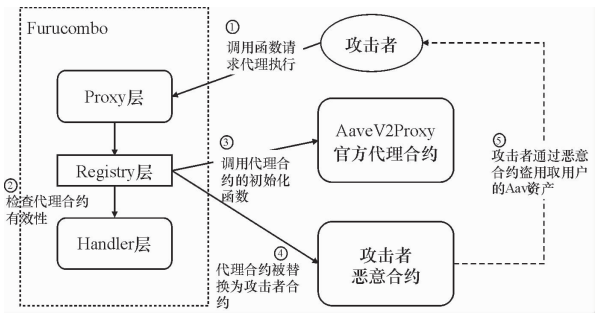

图7 权限攻击流程图

Fig. 7 Permission approve attack flow chart

(1)攻击者调用函数,将官方代理合约AaveV2Proxy地址作为参数传入,请求使用AaveV2模式交易资产;

(2)Furucombo收到请求后,需要检查参数地址有效性,通过后会使用Delegatecall指令调用,且参数可由用户传入,导致用户可以决定具体的函数调用[48];

(3)攻击者传入数据直接调用AaveV2Proxy地址上的初始化函数,该初始化函数只能调用一次,且应该由DeFi官方进行第一次调用,但由于Furucombo未进行初始化,攻击者因此通过该函数将代理合约地址进行了修改;

(4)由于Delegatecall指令[49]中,执行环境为调用者的运行环境,所以攻击者通过初始化函数将Proxy层关于Aave资产的代理合约修改为攻击者恶意合约;

(5)由于Furucombo平台要求用户无限授权,而现在攻击者恶意合约作为平台官方代理合约,拥有关于Aave资产的全部权限,可以直接盗取用户的资产(包括用户未存入Furucombo平台的Aave资产)。

前端攻击:DeFi架构中,通过构建应用程序来与区块链上的DeFi智能合约进行交互[13]。大多数的DeFi平台使用基于Web网络浏览器的前端界面,通过简洁的图形化操作替代复杂的代码来完成各类指令,极大地提高了用户的使用体验及便利性。但是由于使用了常规浏览器的前端界面,使DeFi应用面临了传统Web网络攻击的威胁,包括XSS(cross-site scripting)跨站脚本攻击、CSRF(cross-site request forgery)跨站请求伪造攻击、网络劫持攻击和网络钓鱼。攻击者通过各类Web攻击,控制DeFi的前端网站,在用户进行交易时,将用户的交易对象,从DeFi应用的官方智能合约替换为攻击者的恶意合约,以此盗取资产。

DeFi中的前端攻击主要目的分为2种,第一种是诱导用户调用攻击者的恶意合约,第二种是直接窃取用户的私钥。2种方式中,第一种更为常见,攻击者通过寻找前端代码的漏洞或者使用钓鱼网站,将DeFi前端生成的与官方智能合约交互的交易替换为与恶意合约交互,并诱导用户批准,以此获取交易。前端攻击造成的损失往往是巨大的,用户在不知情的情况下可能会将代币的权限直接授予恶意合约,导致攻击者可以盗取用户的所有资产,而不仅限于盗取该平台上的资产。BadgerDAO[50]以及SushiSwap旗下的MISO应用[51]等都遭受了前端攻击,造成了巨额损失。截至2021年12月,前端攻击至少造成了约1.5亿美元的损失。

前端攻击造成的原因较多,关键在于DeFi应用实际上并未完全的去中心化。DeFi上的协议(即智能合约)实现了去中心化,但是DeFi应用并不能真空运行,仍然需要靠中心化的前端网站和应用程序提供支持,使用中心化的域名、网关进行访问,这就给攻击者提供了使用传统网络攻击方法获利的机会。

2021年12月2日,DeFi应用BadgerDAO收到前端攻击,通过劫持BadgerDAO的前端,诱导用户调用攻击者恶意合约,导致了大量的资产被盗。BadgerDAO应用使用Cloudfare服务作为代理转发,生成交易后请求用户的签署,以此完成交易。在1.1中,介绍了DeFi应用使用web3与区块链智能合约交互的形式(中间通过网桥,将交易包发送给用户的钱包应用请求签名,如果用户同意则交易包上链,完成交易。)详细攻击过程如图

8 所示。

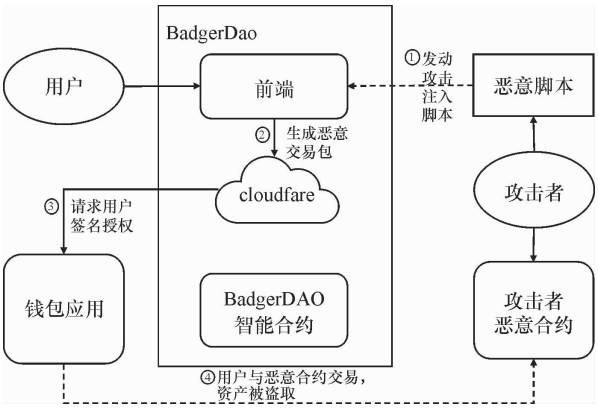

图 8　前端攻击流程图

Fig. 8　Front end attack flow chart

(1)攻击者利用了 Badger 开发人员遗漏的受损 API 密钥,利用了前端的一个漏洞,将恶意脚本注入到前端代码;

(2)用户在 BadgerDAO 网页尝试交易时,被劫持的前端生成与攻击者恶意合约交互的交易包;

(3)通过网桥转发至用户钱包应用,要求用户授予权限,并请求用户签署批准;

(4)用户一旦同意,攻击者则获取了相关资产的代理权限,通过与恶意合约交易,盗取用户的相关所有资产。

2.5　聚合层中的攻击分类

聚合器攻击:DeFi 聚合器是在 DeFi 快速发展到一定规模后才提出的概念。聚合器是 DeFi 应用的进一步延伸,将不同的 DeFi 应用在同一个平台上进行整合,通过一定的算法和规则,尝试获取更大的利益。但是,正是因为需要去聚合各种不同的 DeFi 应用,在各式的 DeFi 协议中间寻找更优的利润,聚合器不可避免地会出现各类的攻击,这类攻击通过寻找聚合器中因为协议兼容性出现的漏洞进行攻击,造成了大量损失。截至 2021 年 12 月,PancakeBunny[52]、PolyYeld Finance[53] 等平台均遭受了攻击,造成了约 4 500 万美元的资产损失。聚合器攻击发生在 DeFi 的聚合层,主要的形式是在不同 DeFi 协议之间交易时,聚合器会铸造代币奖励用户,攻击者利用聚合器漏洞,拉高奖励代币的数量或者价值,以此赚取利益。分析其原因,主要是因为聚合层在提供金融服务时,需要在各类的协议之间交互,而不同的协议因为拥有各自的标准,极易出现标准不兼容的情况,如果开发人员未妥善处理协议间的不兼容性,就会导致出现漏洞,最终造成攻击。2021 年 7 月,聚合器 PancakeBunny 遭受攻击,造成约4 000 万美元的资产损失[52]。Pancake 允许用户抵押流动性代币以此获利,在用户获利时收取一定的费用,并根据收取费用计算一定数量的平台原生代币返回给用户。用户在该平台抵押流动性代币后,可以获取一定的收益,但是由于平台在计算用户收益时,使用了其他 DeFi 的协议获取代币价格用于用户收益计算,导致攻击者利用此漏洞非法获取了大量收益。攻击具体过程如图 9 所示。

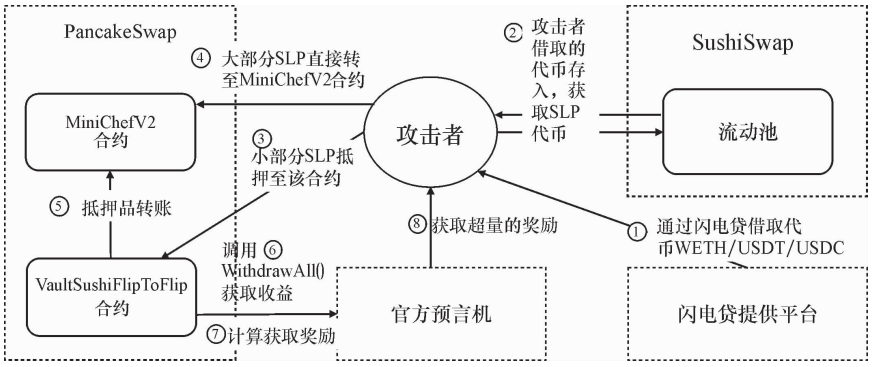

图 9　聚合攻击流程图

Fig. 9　Aggregate attack flow chart

(1)攻击者在 AAVE 闪电贷借出 USDT/USDC/WETH 代币。

(2)攻击者将借取的代币在 DeFi 平台 SushiSwap 中添加流动性获得 SLP。

(3)攻击者将获取的 SLP 小部分抵押到 VaultSushiFlipToFlip 合约。

(4)剩余的大部分 SLP 直接转至 MiniChefV2 合约。

(5)VaultSushiFlipToFlip 合约记录用户的抵押数量,并转至 MiniChefV2 合约。

(6)攻击者调用 VaultSushiFlipToFliph 合约的 WithdrawAll( )函数取出抵押所获取收益,此时,合约会进行铸币,而铸币的 PolyBunny 币由用户抵押品奖励数量和抵押品官方价格共同决定奖励币数量。

(7)聚合器的预言机未被修改,但是抵押品数量被

攻击者操纵。正常的抵押品奖励数量由 VaultSushi-FlipToFliph 合约中计算抵押的占比,再乘以 MiniChefV2 合约中抵押品总量算出。但是攻击者绕过合约记录,直接转账。导致攻击者抵押品在合约中记录的占比正常,但乘以 MiniChefV2 合约中极大的抵押品总数,导致了一个超量的奖励数。

(8)攻击者获取预言机正常价格计算出的奖励代币(如果攻击者正常抵押,此时奖励代币价值应该偏低),并在其他平台卖出此奖励代币后归还闪电贷,以此获利。

3 DeFi 攻击的检测与防御

本节根据每类攻击的特性总结了相对应的检测与防御方法,表 1 总结了 DeFi 攻击的攻击方式以及攻击特征。

3.1 结算层中攻击的检测与防御

结算层中根据是否依赖交易来进行分类,分为了预言机攻击和三明治攻击。表 2 总结了结算层中攻击的检测与防御。

表 1 DeFi 攻击方式以及攻击特征

Table 1 DeFi attack methods and characteristics

攻击所在层次	攻击分类	现有的攻击方式	攻击特征
结算层	三明治攻击	套利机器人	通过抢先交易来完成套利攻击
	预言机攻击	操作喂价源来破坏协议内部的核算	攻击目标依赖 AMM 提供的信息对其内部资产进行定价
资产层	不一致攻击	操作 DeFi 与特定 Token 交互	利用 Token 的特殊性
协议层	重入攻击	利用智能合约存在的重入漏洞	攻击者重复调用平台关键操作来实现获利
	参数校验攻击	利用智能合约缺少校验	攻击者绕过校验来进行攻击
应用层	权限攻击	利用平台的无限授权造成攻击	用户向平台批准代币的无限授权
	前端攻击	XSS、CSRF、网络劫持等传统 Web 攻击	将正常的智能合约交互的交易替换为与恶意合约交互
聚合层	聚合攻击	利用聚合器在多个项目间的询价	聚合器的不合理设计

表 2 结算层中攻击的检测与防御

Table 2 Detection and defense of settlement layer attacks

攻击分类	防御检测方法
预言机攻击	预言机将收集到的价格按交易量加权去除异常值并按照时间同步更新
三明治攻击	零知识证明技术将交易进行加密、AMM 算法优化使用延迟调整的机制

预言机攻击的检测与防御:为了检测套利,并识别出套利攻击行为,现有方法主要是针对 DeFi 的语义进行检测。Wu 等[54]提出了一种基于恢复的 DeFi 语义检测攻击的方法,检测过程如下:

(1)将收集到的以太坊原始交易构造成现金流树(CFT),CFT 用于将原始交易转换为 Token 转移,对该树进行剪枝操作,可以减少绝大部分无用的交互。

(2)定义 DeFi 的行为,并从 CFT 中提升 DeFi 语义。具体行为如下,转移:指将 Token(资产)从一个地址(发送方)转移到另一地址(接收方)。此外,在 ERC20Token 标准定义中,当将支出者字段设置为零地址时,这意味着进行 Token 挖掘,即将 Token 直接存入地址(收件人)中。同样,如果收件人设置为零地址,则表示 Token 正在燃烧;流动性挖矿和流动性取消:为了获得更多的流动性,DeFi 应用程序发行了 LPToken 以激励用户提供流动性(存款加密货币),这被称为流动性挖掘。此外,流动性提供者可以使用 LPToken 作为证书赎回加密货币,这被称为流动性取消。因此,流动性挖掘包括 2 个部分,即存放用户的流动性和铸造 DeFi 应用的 LPToken。取消流动性包括燃烧 LPToken 和赎回已存入的流动性;交易:在正常情况下,交易包括 2 次代币转移。通过流动资金池来交换 Token 通过 2 个启发式准则来进行语义的恢复,如果 CFT 树中相邻的 2 个叶子符合以上定义的 DeFi 行为条件,则将它们合并为 DeFi 高级操作;如果在相邻的 2 片叶子中存在转移链,则将其中一个并入另一个。

(3)通过将恢复的语义与攻击模式进行匹配来检测是否有价格操纵攻击。

Wu 等[54]的工作所覆盖的 DeFi 种类并不全面。本文提出一种方法,通过使用一个去中心化的预言机网络来寻找反映广泛市场覆盖的汇率的真实数值,这样可以

覆盖到所有的 DeFi 应用。DEX 作为交易所是去中心化的,但作为价格参考信息它是中心化的。为了防止收到恶意用户干扰,预言机应收集所有中心化和去中心化交易所的价格,按交易量加权并去除异常值再按照时间同步更新,确保提供给智能合约的数据可靠、可信、抗干扰。提供报价更新的同时维护、调整 AMM 的权重,确保内部汇率与外部市场价格保持匹配,并通过验证机制、异常报警机制等有效拦截攻击者对价格、汇率的操纵,防止套利空间的产生。

三明治攻击的检测与防御:Qin 等[19]根据三明治攻击的特征设定一些启发式规则来检测三明治攻击,但是关于防范三明治攻击并没有提及。本文根据三明治攻击的本质,从如何避免抢先交易方面提出 2 种解决方法:①使用零知识证明技术来将每笔交易的信息都加密隐藏起来,让机器人无从下手。当前的以太坊交易执行机制是通过 Gas 竞争来完成的,即谁出的 Gas 费高,矿工就优先打包谁的交易,如果绕过这种机制,把交易发给矿工让其直接打包,就杜绝了抢跑机器人在中途攻击的可能性。通过构建交易者和矿工之间的桥梁,交易者可以通过打赏的形式让矿工直接打包自己的交易,这就避免了被 抢先交易的可能。②在 AMM 机制下,大额交易产生过大的价格滑点,是抢先交易的利润空间。可以改变 AMM 机制的计算函数曲线,另外采用预言机对交易资产进行报价来降低滑点

3.2　资产层中攻击的检测与防御

资产层的攻击主要是利用了 Token 的特殊性来与 DeFi 进行交互,从而造成攻击。表 3 总结了资产层中攻击的检测与防御。

表 3　资产层中攻击的检测与防御

Table 3　Detection and defense of asset layer attacks

攻击分类	防御检测方法
不一致攻击	通过分析比对 Token、DeFi 账本的记录与 DeFi 的语义是否一致来检测攻击

不一致攻击的检测与防御:现有分析工具可以检查漏洞智能合约中的关系,但很难检测到 DeFi 平台出现的不一致行为,因为这类检测工具专注于代码漏洞而不是设计缺陷,比如重入漏洞、整数溢出、没有检查的 call 调用等。案例中的 DeFi 应用没有检查代币是否实际收到,导致假存款。实际上,这是一个设计缺陷而不是代码漏洞。

本文提出一种基于模式匹配的方式来检测不一致攻击。检测原理如图 10 所示。

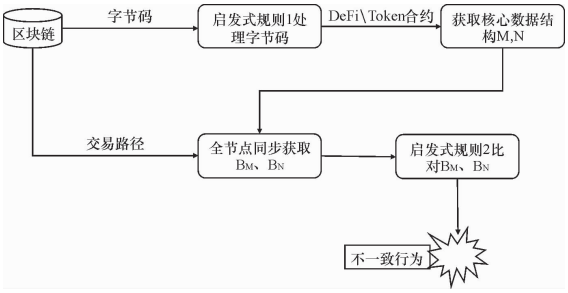

图 10　不一致检测方法示意图

Fig. 10　Schematic of inconsistent behavior detection

(1)通过启发式规则从字节码中筛选出 DeFi 以及 Token 合约来获取,启发式规则 1 是字节码符合 ERC - 20 和 ERC - 777 标准的 Token 合约,以及字节码中含有 DeFi 的特殊函数,比如 swap( ),deposit( ),withdraw( );

(2)通过符号执行来识别出 DeFi 和 Token 的核心数据结构的字节码序列;

(3)拿到 M、N 后通过以太坊客户端 Geth 的全节点同步获取每笔交易中的 Bm,Bn;

(4)通过启发式规则 2 来比对交易行为,启发式规则 2 是在一笔完整的交易路径中改变了 M、N 并且 Bm、Bn 的不匹配则被判定为不一致。

DeFi 上的智能合约在处理兑换逻辑过程中,需要检查进行兑换的 2 种代币在兑换过程中合约是否收到了相应的代币,保证代币的余额正确记录,不能只是依赖 ERC20 等 Token 标准中关于转账的返回值,从而避免因代币余额记录错误导致的问题。

3.3　协议层中攻击的检测与防御

协议层中的攻击根据 DeFi 智能合约的漏洞来进行分类,分为重入攻击和参数校验攻击。表 4 总结了协议层中攻击的检测与防御。

表 4　协议层中攻击的检测与防御

Table 4　Detection and defense of protocol layer attacks

攻击分类	防御检测方法
重入攻击	修改漏洞检测工具检查条件,设置外部代币合约白名单
参数校验攻击	对敏感函数做严格的权限检查以及智能合约检测工具来预防攻击

重入攻击的检测与防御:DeFi 重入攻击来源于智能合约的重入漏洞,现有检测防御方法都是在智能合约漏洞的基础上进行的:①使用如 Oyente, ZEUS, Vandal, Ethir 等[28]检测工具对智能合约代码进行检测,挖掘可能存在的重入漏洞。使用工具分析的方法存在误报率高且效率较低的问题。②增加修饰器,对关键变量进行加锁操作。通过限制关键操作,防止在攻击者通过外部

调用,使合约中的关键函数被执行多次。

本文基于DeFi平台重入攻击的特点,提出新的解决方法。首先分析DeFi平台重入攻击需要的3个条件:①调用关键函数前是否有修饰器或者变量用于限制关键操作;②关键函数中的转账操作是否在修改平台资产记录操作之前进行;③平台是否允许使用用户自定义的资产合约。

基于此条件,解决方法如下:①优化现有检测工具。修改漏洞检测工具检查条件,可以预先定义DeFi平台的各类关键操作和变量,检测工具需要检测是否有修饰器限制重入路径,且检测工具只有在检测到外部调用后又修改了关键函数才算作DeFi智能合约的重入漏洞;②平台可以设置外部代币合约白名单,只有符合ERC标准的外部代币合约才加入白名单。白名单内的合约可以正常建立流动池,对于不在白名单内的外部合约调用,应该保证不对DeFi平台的关键变量进行修改。

参数校验攻击的检测与防御:此类攻击多数都是因为合约代码本身的设计缺陷,现有的检测工具smartcheck[55]可以对返回值以及交互中的数据没有进行校验的智能合约进行检测,但是存在检测效率低以及漏洞检测不完全等问题。

为了防止DiFi的参数校验攻击,本文提出如下解决方案:①所在合约部署的主网之前做好外部审计,另外可以通过一些测试套件的样本作为参考的起点,如chainlink的套件仓库[56],来提供一种检测思路。②所有由用户调用的函数都要对传入的参数进行合理性检查,避免参数使用不合理导致的异常。在使用具有限制的函数时,要验证传入不合要求的参数是否会绕过限制执行,或者有其他类似的函数可以进行绕过。③对铸币、权限更改等敏感函数做严格的权限检查,并根据业务逻辑确定这类函数的可见性。

3.4 应用层中攻击的检测与防御

应用层的攻击根据是否为传统的Web攻击进行分类,分为了权限攻击和前端攻击。表5总结了应用层中攻击的检测与防御。

表5 应用层中攻击的检测与防御

Table 5 Detection and defense of application layer attacks

攻击分类	防御检测方法
权限攻击	使用热钱包、代理合约、EIP-2612标准来预防攻击
前端攻击	应用与智能合约捆绑签名,应用通过签名来匹配

权限攻击的检测与防御:现有检测防御方法有:①使用权限检测工具如Token Allowance Checker[57]、smart contract allowance checker[58]等工具检测用户在各平台的授权状况,并可以对授权行为进行撤销或修改。这类工具通过重放以太坊交易的模式,检测用户在代币资产上的Approve()函数调用情况,获取用户授权额度,并通过重新调用代币资产的权限函数修改授权额度,但该方法只能减少用户资产被盗的风险,不能解决权限攻击问题。②DeFi平台限定授权额度。每次交易时,用户只向DeFi平台授权相应额度的权限,而不是无限授权的方式。如Zapper等[59] DeFi平台都可以让用户自由选择是否要限制授权。但每次授权时,平台都需要额外发起一笔交易,带来了新的费用,给用户带来了不便。

为了防止DeFi的权限攻击,平衡便利性和安全性之间的关系,本文提供多种方案解决DeFi平台的授权问题:①使用热钱包。如果要使用DeFi平台的无限授权功能,可以借助热钱包,将需要在该平台的交易资产单独划分到另一个独立地址。这样,即使遭受安全攻击,也只有该部分的资产收到威胁,降低了用户使用无限授权功能的风险。②使用代理合约。该方法较为复杂,且有一定的技术门槛,要求用户自己编写一个代理合约,使用此代理合约直接与DeFi平台合约进行交互。用户向代理合约可以提交无限授权,而代理合约与DeFi平台交互时,会自动批准单笔交易所需要的额度,再执行交换交易,并由代理合约接收代币。③使用EIP-2612标准。在该标准下,用户可以通过发送消息的方式来赋予平台每笔交易权限,而不是发送交易的方式。因此,可以节约每次赋权所需的额外费用。但由于该标准下的开发生态一般,需要更多的DeFi项目逐步迁移使用该标准。

前端攻击的检测与防御:由于前端攻击较多来源于传统网络攻击方式,现有的解决方法有:①直接使用Web3官方接口与DeFi智能合约交互。但该方法需要用户自己编写代码,具有较高的门槛,用户体验差,较少被使用。②加强Web网络安全防御。如使用安全套件编写前端代码,以及仅执行安全域名下的脚本等方法。③Uniswap提出使用IPFS(InterPlanetary File System)[60],即星际文件系统技术,将前端代码分割到网络节点中,使用IPFS服务在用户访问时再获取前端代码,防止前端被劫持。④DeFi应用Liquity使用ICP[61]服务,通过ICP的DNS服务器为每个域名指定各自的web文件,以此将域名和前端内容绑定,同时记录前端代码修改记录,防止前端页面被劫持。

针对DeFi应用的特点,本文提出以下解决方案:为了保证DeFi应用前端没有更改,防止攻击者欺骗用户与恶意合约的交易,将DeFi官方智能合约地址与前端应用捆绑,并将该捆绑包进行签名,然后用户在使用钱包应用签署交易时,钱包应用会检查前端应用程序签

名,若接收到的前端包的计算签名与签名地址不匹配,则会显示警告消息,并建议用户不要与应用程序交互,通过方法减少钓鱼,劫持等前端攻击的风险。

3.5　聚合层中攻击的检测与防御

聚合层中的攻击主要是聚合器合约在计算其他 DeFi 项目的数据上面设计不合理。表 6 总结了聚合层中攻击的检测与防御。

表 6　聚合层中攻击的检测与防御

Table 6　Detection and defense of aggregation layer attack

攻击分类	防御检测方法
聚合攻击	多协议审计、延迟预言机、使用多个协议对价格进行验证

聚合器攻击的检测与防御:由于 DeFi 聚合器发展时间较短,在 DeFi 应用兴起后,才有对各种 DeFi 应用进行综合整理的聚合平台,导致缺少专门针对聚合层攻击的检测和防御方法。本文在其他架构层的检测防御方法基础上,提出以下新的方案:①加强兼容性协议审计,聚合器平台应对需要交互的 DeFi 协议和相关接口进行严格审计。审计内容主要是函数或交互接口,以及不同协议的价格或收益计算方法等。②使用延时的价格预测机制,防止聚合器在不同协议间交互时,平台的价格机制被恶意用户操控。使用延迟调整的机制来减少大额交易对后续交易价格的影响,可以有效防止抢先交易攻击。比如当发生兑换交易时,交易池价格不会立刻调整成真实价格,而是在若干分钟内,缓慢的趋向真实价格。③在涉及到使用其他 DeFi 合约进行价格计算时,可以使用多个 DeFi 合约价格进行综合验算,减小因为一个 DeFi 协议价格被操控造成的风险。

4　DeFi 攻击的未来研究中重点

4.1　组合型攻击的研究

随着去中心化金融市场的发展,DeFi 风险成为焦点,从之前的重入攻击泛滥,到现在的闪电贷攻击,都是因为加密资产可以以重复抵押的方式在 DeFi 中进行交换,这样项目方会面临着组合性攻击。研究者未来可以针对现有的攻击而形成新的检测攻击的策略。

4.2　矿工可提取价值的研究

如今看到的大多数矿工可提取价值(MEV: Miner Extractable Value) 形式都不是来自矿工本身,而是来自第三方机器人。这些机器人通过改变他们支付给矿工的交易费用来操纵他们在一个区块内的交易顺序。这意味着即使矿工根据最高 gas 价格订购交易,也可以提取 MEV。然而,MEV 可以被视为矿工可以提取多少价值的上限,因为矿工可以利用他们的特权地位进行抢先交易中从而获利。因此,如何解决公平的排序服务是未来的挑战之一。

4.3　漏洞检测工具的研究

目前大量的工作都用来分析智能合约的现有漏洞,当前的检测分为动态和静态检测 2 种,这 2 种并没有关注智能合约的可组合性,使得这些工具无法推断因智能合约外部变化而出现问题的场景,比如预言机返回的价格突然改变以及对于复杂场景交互下的路径爆炸问题。此外,大多数工具很少对智能合约的语义属性进行推理,例如特定执行路径如何影响 ERC-20 代币余额未来这些领域的改进将使审计人员和开发人员能够更有信心地分析和部署他们的合同,从而减少技术安全漏洞的数量。

5　总　结

DeFi 的攻击方式多种多样,攻击与攻击之间是有耦合性的,单独的给每种攻击方式进行分类只会体现每个攻击之间的差异性。本文以 DeFi 的分层架构将 DeFi 的攻击进行分类,分成了 5 大类:聚合层的攻击、应用层的攻击、协议层的攻击、资产层的攻击与结算层的攻击. 其中聚合攻击属于聚合层;应用层攻击分为权限攻击和前端攻击;协议层分为重入攻击和参数校验攻击;不一致攻击属于资产层;结算层分为三明治攻击和预言机攻击。本文在对攻击进行分类的同时,还介绍了对各类型攻击的检测与防御方法。通过对 DeFi 攻击进行层次分类有助于各项目方在实现底层算法时进行审核,从而促进 DeFi 平台的稳定发展。

参考文献:

[1] Chen T, Li Z, Zhu Y, et al. Understanding ethereum via graph analysis[J]. ACM Transactions on Internet Technology (TOIT), 2020, 20(2):1-32.

[2] Wang B, Liu H, Liu C, et al. Blockeye: Hunting for DeFi attacks on blockchain[C]//2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). Piscataway:IEEE, 2021: 17-20.

[3] Schar F. Decentralized finance: On blockchain-and smart contractbased financial markets[J]. Federal Reserve Bank of St. Louis Review, 2021(2):153-174.

[4] Elliptic. DeFi: Risk, regulation, and the rise of decrime[EB/OL]. (2021-11-18)[2021-12-20]. https://www.elliptic.co/resources/DeFi-risk-regulation-and-the-rise-of-decrime.

[5] Wayn. Tokeninsight 2021 crypto yearly review[R]. TokenInsigh,2021.
[6] AFP. Record cryptocurrency heist valued at 600 million[EB/OL]. (2021-08-10)[2021-11-26]. https://www.security-week.com/record-cryptocurrency-heist-valued-600-million.
[7] Tiwari A. Wormhole hack illustrates danger of DeFi cross-chain bridges[EB/OL]. (2022-02-16)[2022-02-20]. https://cointelegraph.com/news/wormhole-hack-illustrates-danger-of-DeFi-cross-chain-bridges.
[8] Patrick T. The DeFi hack of 2020[EB/OL]. (2021-11-18)[2022-01-02]. https://coingeek.com/the-DeFi-hacks-of-2020/.
[9] Qin K, Zhou L, Gervais A. Quantifying blockchain extractable value: How dark is the forest? [EB/OL]. (2021-12-10)[2022-02-05]. http://arXiv preprint arXiv:2101.05511,2021.
[10] Wang S H, Wu C C, Liang Y C, et al. Promutator: Detecting vulnerable price oracles in DeFi by mutated transactions[C]// 2021 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). Piscataway: IEEE, 2021: 380-385.
[11] Tolmach P, Li Y, Lin S W, et al. Formal analysis of composable DeFi protocols[C]//International Conference on Financial Cryptography and Data Security. Ithaca: Cornell University, 2021: 149-161.
[12] Jensen J R, Von Wachter V, Ross O. An introduction to decentralized finance (DeFi)[J]. Complex Systems Informatics and Modeling Quarterly, 2021(26):46-54.
[13] Katona T. Decentralized finance: The possibilities of a blockchain "money lego" system[J]. Financial and Economic Review, 2021, 20(1):74-102.
[14] Werner S M, Perez D, Gudgeon L, et al. Sok: Decentralized finance (DeFi)[EB/OL]. (2021-09-26)[2022-01-20]. http://arXiv preprint arXiv:2101.08778, 2021.
[15] Kim J, Kim S. A survey of decentralized finance (DeFi) based on blockchain[J]. Journal of the Korea Society of Computer and Information,2021, 26(3):59-67.
[16] Xu T A, Xu J. A short survey on business models of decentralized finance (DeFi) protocols[EB/OL]. (2022-02-15)[2022-02-21]. http://arXiv preprint arXiv:2202.07742, 2022.
[17] Corbet S, Goodell J W, Gunay S, et al. Are DeFi tokens a separate asset class from conventional cryptocurrencies? [EB/OL]. (2021-04-20)[2022-01-13]. Available at SSRN 3810599, 2021.
[18] Popescu A D. Transitions and concepts within decentralized finance (DeFi) space[J]. Research Terminals in the Social Sciences, 2020.
[19] Qin K, Zhou L, Afonin Y, et al. Cefi vs. DeFi. comparing centralized to decentralized finance[EB/OL]. (2021-06-16)[2021-12-27]. http://arXiv:2106.08157, 2021.
[20] Yli-huumo J, Ko D, Choi S, et al. Where is current research on blockchain technology? ——A systematic review[EB/OL]. (2016-10-03)[2021-12-13]. http://doi.org/10.1371/journal.pone.0163477.
[21] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[J]. Decentralized Business Review, 2008:21260.
[22] Wood G. Ethereum: A secure decentralised generalised transaction ledger[R]. Ethereum, 2018.
[23] Richards S. Gas and fees[EB/OL]. (2021-11-12)[2022-01-02]. https://ethereum.org/en/developers/docs/gas/.
[24] Xu J, Paruch K, Cousaert S, et al. Sok: Decentralized exchanges (dex) with automated market maker (amm) protocols[EB/OL]. (2022-01-14)[2022-02-10]. http://arXiv preprint arXiv:2103.12732, 2021.
[25] Engel D, Herlihy M. Composing networks of automated market makers[C]//Proceedings of the 3rd ACM Conference on Advances in Financial Technologies. Ithaca: Cornell University,2021: 15-28.
[26] Ni Y D, Zhang C, Yin T T. Smart contract security vulnerability research[J]. Journal of Cyber Security, 2020, 5(3):78-99.
[27] Chen T, Zhang Y, Li Z, et al. Tokenscope: Automatically detecting inconsistent behaviors of cryptocurrency tokens in ethereum[C]//Proceedings of the 2019 ACM SIGSAC conference on computer and communications security, New York: Association for Computing Machinery, 2019: 1503-1520.
[28] Praitheeshan P, Pan L, Yu J, et al. Security analysis methods on ethereum smart contract vulnerabilities: A survey[EB/OL]. (2020-09-16)[2021-12-20]. http://arXiv preprint arXiv:1908.08605, 2019.
[29] Finance B. DeFi security lecture 4 . front-end attack[EB/OL]. (2021-10-16)[2022-01-23]. https://medium.com/beaver-finance/defi-security-lecture-4-front-endattack-44f32ca0cd68.
[30] Chainlink. What is a blockchain oracle? [EB/OL]. (2021-09-14)[2022-02-07]. https://zh.chain.link/education/blockchain-oracles.
[31] 曹一新. DeFi 经济攻击的一般模式分析[EB/OL]. (2021-03-10)[2022-02-07]. https://cj.sina.com.cn/articles/view/6311913111/178382697020013p92.
[32] Slowmist. 精析 DeFi 协议 Warp Finance"预言机"攻击事件[EB/OL]. (2021-12-24)[2022-01-03]. https://bc.cnvd.org.cn/bulletin_info? num = a08d2c81371e26e8189bdca5c0491d71.
[33] Dzyatkovskii A. No sandwich, please! popular DeFi attack strategy analysis[EB/OL]. (2021-05-27)[2021-12-19]. https://hackernoon.com/no-sandwich-pleasepopular-DeFi-attack-strategy-analysis-jk1734rf.
[34] Degate. Analyze the principle and solution of ethereum's preemptive transaction[EB/OL]. (2021-07-08)[2021-12-28].

https://www.sohu.com/a/476251523_100217347.
[35] 1Inch Network. Balancer pool with sta deflationary token incident[EB/OL]. (2021-06-29)[2021-09-29]. https://blog.1inch.io/balancer-hack-2020-a8f7131c980e.
[36] Ethereum. Security considerations[EB/OL]. (2021-09-13)[2021-11-15]. https://docs.soliditylang.org/en/latest/security-considerations.html.
[37] Gudgeon L, Perez D, Harz D, et al. The decentralized financial crisis[C]//2020 Crypto Valley Conference on Blockchain Technology (CVCBT). Piscataway: IEEE, 2020: 1-15.
[38] Apr T. imbtc uniswap pool drained for $ 300k in eth[EB/OL]. (2020-04-18)[2021-11-16]. https://defirate.com/imbtc-uniswap-hack/.
[39] Cimpanu C. Hacker steals $ 2 million from cryptocurrency service akropolis[EB/OL]. (2020-11-13)[2021-12-12]. https://www.zdnet.com/article/hacker-steals-2-million-from-cryptocurrency-service-akropolis/.
[40] Behnke R. Explained: The burgerswap hack[EB/OL]. (2021-06-02)[2021-10-27]. https://halborn.com/explained-the-burgerswap-hack-may-2021/.
[41] Lian'AN. Basics of DeFi security issues[EB/OL]. (2021-12-28)[2021-12-29]. https://www.defidaonews.com/media/6718708.
[42] Craig T. Monox finance drained of $ 31m in latest DeFi hackk[EB/OL]. (2021-11-30)[2021-12-23]. https://cryptobriefing.com/monox-finance-drained-of-31m-in-latest-defi-hack/.
[43] Team D S. Disclosing a recently discovered exchange vulnerability[EB/OL]. (2020-06-14)[2021-10-03]. https://defirate.com/defi-saver/.
[44] Behnke R. Explained: The superfluid hack[EB/OL]. (2022-02-01)[2022-02-06]. https://halborn.com/explained-the-superfluid-hack-february-2022/.
[45] Rahimian R, Eskandari S, Clark J. Resolving the multiple withdrawal attack on erc20 tokens[C]//2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW). Ithaca: Cornell University, 2019:320-329.
[46] Rekt. Explained: The furucombo evil contract hack[EB/OL]. (2021-02-27)[2021-12-06]. https://halborn.com/explained-the-furucombo-evil-contract-hack-feb-2021/.
[47] Eimantas. Unlimited erc20 allowances considered harmful[EB/OL]. (2021-10-07)[2021-12-06]. https://kalis.me/unlimited-erc20-allowances/.
[48] Wang A, Wang H, Jiang B, et al. Artemis: An improved smart contract verification tool for vulnerability detection[C]//2020 7th International Conference on Dependable Systems and Their Applications (DSA). Piscataway: IEEE, 2020: 173-181.
[49] Ethereum. Dark mode delegatecall[EB/OL]. (2020-10-09)[2021-12-06]. https://solidity-by-example.org/delegatecall/.
[50] Waldman A. Badgerdao users' cryptocurrency stolen in cyber attack[EB/OL]. (2021-10-07)[2022-01-06]. https://www.techtarget.com/searchsecurity/news/252510627/BadgerDAO-users-cryptocurrency-stolen-in-cyber-attack.
[51] Mandal R. Sushiswap's miso launchpad loses $ 3 million in an attack[EB/OL]. (2021-09-18)[2021-11-15]. https://www.cryptotimes.io/sushiswaps-miso-launchpad-loses-3-million-in-an-attack/.
[52] Jakobson L. The tragicomedy of pancakebunny[EB/OL]. (2021-10-13)[2021-11-17]. https://coinmarketcap.com/alexandria/article/the-tragicomedy-of-pancakebunny.
[53] PolyYeld Finance. DeFi project on polygon, polyyeld, was hacked[EB/OL]. (2021-08-01)[2021-11-03]. https://info1.net/defi-project-on-polygon-polyyeld-was-hacked-yeld-token-price-plummets-202111876/.
[54] Wu S, Wang D, He J, et al. Defiranger: Detecting price manipulation attacks on DeFi applications[EB/OL]. (2021-04-20)[2021-11-29]. http://arXiv preprint arXiv:2104.15068, 2021.
[55] Tikhomirov S, Voskresenskaya E, Ivanitskiy I,et al. Smartcheck: Static analysis of ethereum smart contracts[C]//Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain. Piscataway: IEEE, 2018: 9-16.
[56] Chainlink. Chainlink developer starter kit demonstration[EB/OL]. (2021-05-07)[2021-11-03]. https://blog.chain.link/starter-kit-showcase-zh/.
[57] Kalis R. How to protect your tokens from DeFi attacks[EB/OL]. (2020-02-21)[2021-04-02]. https://cryptonews.com/news/how-to-protect-your-tokens-from-defi-attacks-5849.htm.
[58] Unrekt. Smart contract allowance checker[EB/OL]. (2020-10-16)[2021-10-05]. https://app.unrekt.net/.
[59] Zapper. Set allowance if needed[EB/OL]. (2021-01-02)[2021-10-05]. https://docs.zapper.fi/zapper-api/api-guides/exchange-assets#set-allowance-if-needed.
[60] Chauhan A. A complete guide to building ethereum dapps: Frontend and back-end[EB/OL]. (2021-01-18)[2022-01-20]. https://betterprogramming.pub/a-complete-guide-to-build-ethereum-dapps-front-end-and-back-end-6fa44b66554b.
[61] Yolo. Look at the two options dapp how to deal with the risk of front end hosting[EB/OL]. (2021-06-20)[2022-01-26]. https://cdmana.com/2021/06/20210620050525221J.html#.

【责任编辑:陈　钢】