区块链技术与数据经济学 A

【USENIX Security 21】研究成果概括

姓名：	曹卓文
学号：	19121442
学校：	上海大学
专业：	计算机科学与技术
方向：	智能合约
主页：	github.com/yuban00018

2022 年 5 月 15 日

摘要

本文对 USENIX Security 2021 论文 Evil Under the Sun: Understanding and Discovering Attacks on Ethereum Decentralized Applications[1] 进行了解读。

在该研究中，研究者提出了首个区块链上的数字取证方法 DEFIER，分析衡量现实中对以太坊 Dapp 的攻击。利用在以太坊链上的交易记录，取证攻击痕迹并用以分析，并对 Dapp 的攻击行为总结出了普适性的生命周期并发现了一种多角色协同工作的层次化攻击结构，这些在之前的研究中从未被报道过。

介于该研究首次针对智能合约实施的网络犯罪生态进行了细致的研究，笔者认为介绍这篇文章对帮助研究人员进一步认识智能合约有关的网络犯罪问题有重要参考意义。

本文对该论文的背景、研究方法、结论进行了讨论，最后结合该研究对智能合约与 Dapp 安全的未来进行了探讨。

论文地址:https://www.usenix.org/conference/usenixsecurity21/presentation/su

源码地址:https://drive.google.com/drive/folders/1cdD1gHNbWIS228QXmeUReougSL_k1kvf

论文作者：Liya Su, Indiana University Bloomington

关键词：机器学习，去中心化应用，以太坊，智能合约

1 研究背景及相关工作

分布式应用（Decentralized Application, Dapp）是一种运行在区块链上的应用程序，可以被看作使用智能合约代替传统后端服务器，让服务器运行在区块链网络上的新型应用。随着区块链技术的持续走热，针对智能合约的新型网络犯罪越来越多。其中最著名的便是在 2016 年发生的以太坊 The DAO 事件，共计造成近 5000 万美元的损失 [2]，并导致以太坊发生硬分叉。表1展示了从 2016 到 2019 年发生的针对以太坊 Dapp 的攻击事件分类及其发生次数。

攻击类型	定义	事件发生数
可预测的随机处理	伪随机数被攻击者用以作弊（如赌博游戏）	6
拒绝服务攻击	通过耗尽区块的 gas 或利用合约逻辑漏洞使得合约不能正常运行	4
整数溢出	利用错误的数学表达式导致整型溢出，得到错误值欺骗合约的逻辑判断	26
递归漏洞	合约调用外部合约，外部合约在回调调用合约，使得在不一致的内部条件运行	2
不恰当的授权	利用 Dapp 的身份验证过程来强制执行工作流或访问变量	15

表 1: 针对以太坊 Dapp 的攻击事件

由于智能合约是 Dapp 的基石，针对智能合约的安全问题正引起越来越多的研究人员的关注，Chen 团队 [3] 构建了基于机器学习的庞氏骗局检测工具。Atezi 等人 [4] 对针对以太坊智能合约的攻击进行了调查，分类并详细的讨论了漏洞。Rouhani 等人 [5] 总结归纳了智能合约的安全问题分类，安全分析工具（这些工具主要集中在通过反编译手段来分析合约的程序逻辑漏洞）。以上的研究都侧重于对漏洞的评估，缺乏对该类网络犯罪的生态如攻击生命周期、基础设施以及组织行动进行讨论。

因此，在该研究中，研究者提出了首个基于时序分析的区块链数字取证方法，分析现实中对以太坊 Dapp 的攻击。利用以太坊链中的交易记录的时序数据取证攻击痕迹，并使用长短期神经网络（Long Short-Term Memory, LSTM）分析攻击阶段。

基于时序分析的类似工作有 Milajerdi 等人 [6] 的 HOLMES，这是一个实时的 APT 检测系统，能够总结攻击者的杀伤链，以基于频率分析识别与已知攻击相关的行为。与该工作不同，以太坊 Dapp 攻击领域中的杀伤链及其相关攻击操

作没有得到充分的探索，这与传统的 APT 杀伤链有很大的不同。在该研究中，首次利用基于图序列挖掘的以太坊交易时间序列分析，学习高层攻击意图，使安全人员能够准确地检测出已知和未知的攻击。

2 研究结果

2.1 漏洞利用阶段的分析

该研究采用数据收集和推导方法重建了 42 个现实发生的 Dapp 攻击事件（数据的采集方法如图1所示），包括 126 个语义相似的事务簇和 58,555 个事务。基于这些事务聚类，研究者对它们进行了进一步研究以了解 Dapp 攻击的犯罪足迹。

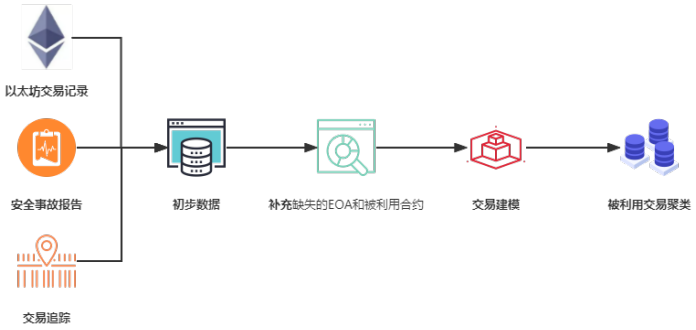


图 1: 数据收集

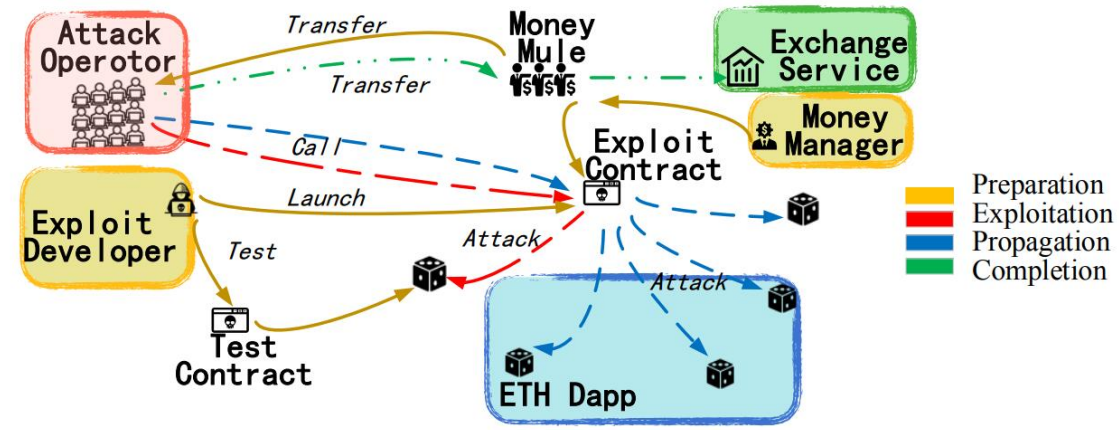
如图2所示，Dapp 攻击可以被划分为以下四个阶段：攻击准备，漏洞利用，攻击传播和完成阶段。

- 在准备阶段，攻击者使用少量的交易对攻击代码进行测试，与此同时，可以观察到资金管理者支付攻击成本（gas 或门票费）到恶意合约中。这些交易通过钱骡（洗钱账户）来隐藏资金管理者的真实账户。这一阶段的 78% 的交易都是用来测试恶意合约的，其中 79% 的被测试 Dapp 在利用或传播阶段被攻击。

- 在利用阶段，多个攻击者使用不同的外部账户唤醒恶意合约来攻击被害 Dapp 并以此获利。在攻击过程中，攻击者倾向于快速发展其策略，通过修改合约或创建新的合约以提高效率（获取更多的收入或支付更少的成本）。

- 在传播阶段，攻击者会尝试使用这类恶意合约来攻击有类似漏洞的 Dapp 以期获得更多的利益。同时这些攻击代码大部分直接从旧的恶意合约复制而来，并且攻击者会扫描存在漏洞相同函数名的 Dapp。

- 在完成阶段，攻击者销毁恶意合约并回收攻击收益，最后这些收益会通过钱骡进行洗钱来混淆痕迹。并且，攻击者倾向于将以太坊代币通过去中心化交易所转换为以太币。



Example of Dapp criminal footprints.

图 2: Dapp 犯罪足迹

笔者认为，通过以上对攻击阶段的总结，不难得出当前针对 Dapp 的攻击呈现出分工化和流程化趋势，但这也代表着针对 Dapp 的攻击是有迹可循的，为作者接下来利用时序数据来探测未知的攻击带来了可能。

2.2 DEFIER 的设计理念

针对以上的发现，研究人员设计了 DEFIER（Dapp Exploit Investigator），由两部分组件组成：预处理和基于时序的分类。

- 预处理首先将于 Dapp 直接交互的一组交易作为输入，并自动扩展该集合以纳入和 Dapp 间接相关的其他交易。然后这些交易根据执行轨迹的相似性和调用时间的接近程度（在一个较短的窗口中）进行聚类。

- 基于时序的分类对每个交易序列，即由合约和 EOA 组成的带权有向图通过嵌入（embedding）建模为特征向量，该研究提出了一种新的嵌入方法，将交易序列转换为捕获序列潜在意图的特征向量（使用注意力模型 [7] 来关注每个交易和 Dapp 之间的互动以及交易之间的关系）。接着这些向量会经过多类分类器，对于存在攻击意图的则输出所属攻击阶段。

2.2.1 预处理的扩容、降噪和聚合

通过先前对 Dapp 攻击的调查可以发现，虽然直接对 Dapp 的攻击的恶意合约是显而易见的，但实际上针对 Dapp 的攻击存在一条完整的杀伤链，杀伤链有多种角色组成，这些 EOA 可能不直接和 Dapp 交互。因此该研究提出了一种规则来发现所有相关 EOA，其具体实现如下。

首先识别所有与 Dapp 直接交互的 EOA，包括直接调用 Dapp 的地址和创建合同调用的地址。然后，给定一个通过交易 tx_s 与 Dapp 交互的合约 S，收集所有创建、调用或转移资金到合约 S 的 EOA。通过以上方式发现所有相关的 EOA，并用交易来描述每个 EOA 的行为。利用这样的一个算法，我们可以轻易地发现和 Dapp 攻击有关的所有交易，伪代码描述见算法1。

Algorithm 1: 扩展相关交易算法

```
Data: Dapp: Dapp 及其地址
EOAs = extract_eoa_of_dapp(Dapp);
interval = 1 day;
threshold = 3;
for EOA∈ EOAs do
    /* 对于和 Dapp 发生交易的合约 S，找出合约 S 的所有关联交易
       txs                                                    */
    txs = get_txs_by_Dapp_and_Eoa(Dapp,EOA);
    for tx∈ txs do
        date = tx_date(tx);
        /* 生成关注区间                                        */
        focus_period = calculate_period(date,interval);
        /* 获得某个账户在此期间的交易行为                      */
        extend_txs = get_tx_in_period(EOA,focus_period);
        /* 选取该账户在关注区间的，与 tx 类似的交易行为 etx    */
        picked_txs = [etx for etx in extend_txs if distance(tx,etx)≤ threshold];
        /* 保存存在关联的新交易                                */
        save(picked_txs);
    end
end
```

为了降低噪声（Dapp 拥有方的 EOA 或者是管理用户信息的库合约），研究人员分析调用方（排除 Dapp 主动调用的合约）以及对合约字节码进行反汇编，

通过正则表达式寻找库合约等方式，排除了和攻击调查无关的 EOA。最后研究人员对具有相似执行轨迹，或者在近似时间段中发生的交易进行聚类。

2.2.2 基于序列的分类模型

为了从交易中提取信息，研究人员从每个交易聚类中形成一个交易序列，并按其时间戳排序。对于一个交易序列，根据具有类似语义的其他序列的知识，预测其潜在意图（例如，利用测试、攻击传播等）来确定该交易序列是否代表了对一个 Dapp 的攻击。

与 Dapp 攻击阶段有关的相似语义交易序列 $\hat{s}$ 可以被表示为一个二元组 $(tx_i|i=1\ldots k, y)$，其中 $tx_i|i=1\ldots k$ 是序列 $\hat{s}$ 中的交易，y 是攻击阶段的标签。基于时序的分类目标就是在分类器的模型参数 θ 下为输入序列 $\hat{s}$ 找到标签 y。其中参数 θ 通过训练数据集学习。序列 s 被送入一个 LSTM 模型，生成一个向量 h，描述交易之间的关系，并突出与恶意行为有关的信息。

研究人员选择了 Bi-LSTM，一种改良的 RNN，因为它被设计用来学习序列中元素之间的长期依赖关系 [8]，这对于识别在不同攻击阶段将交易联系在一起的模式至关重要。如图3所示，研究人员使用了 EOA-Dapp-execution 注意力模型来突出 EOA 对于 Dapp 的目的信息。注意力 a_i 被用来调整交易图 tg_i 的向量表示，它由 eoa_i(EOA),d_i（Dapp，由顶点嵌入产生 [9]）,tg_i（由图嵌入产生）的向量表示加权组合决定的。其权重又是通过 Bi-LSTM 模型学习的，最终输出描述输入的特征向量 h，即交易序列的目的。

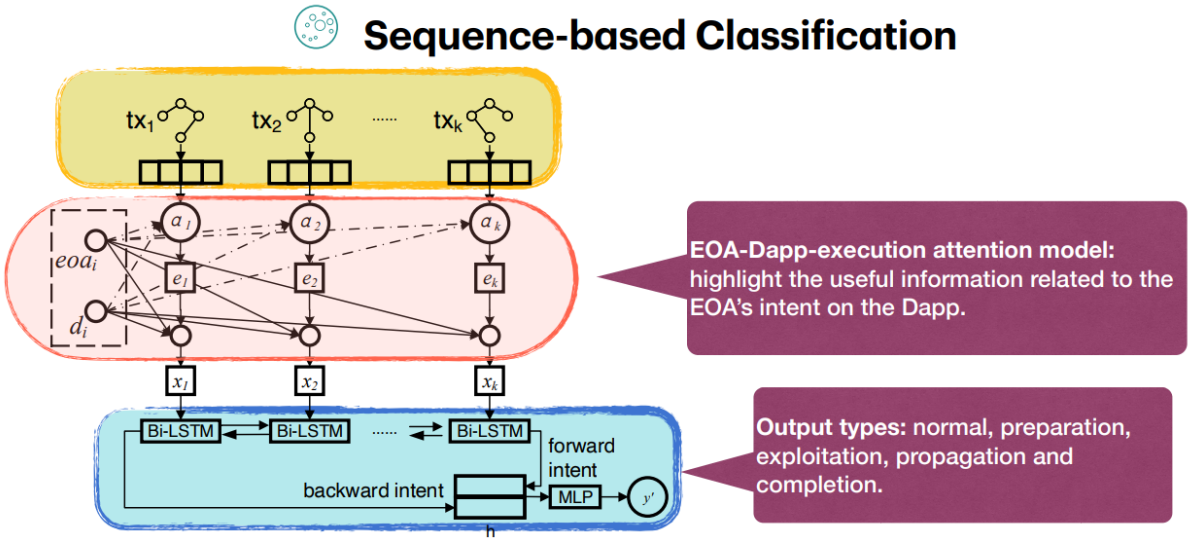


图 3: 基于序列的分类

而注意力模型输出的特征向量 h 将会作为多层感知机（Multilayer Perception,

MLP）的输入，DEFIER 使用 MLP 来生成序列对于给定的攻击阶段的相关概率 y'。该研究中，研究者建立了三层的 Bi-LSTM 网络，其卷积大小为 128，隐藏状态维数大小为 256，批处理大小为 128，epochs 设置为 20，学习率设置为 0.0001。

2.3 实验结果和发现

研究人员从 Dapp 榜单 [10] 采集了 104 个活跃的 Dapps 以及他们的合约地址。使用 DEFIER 的预处理器汇集了 2,350,779 次交易，并构建了 342,224 个交易聚类。DEFIER 检查了这些交易并标注了其中的 476,342 次交易（涵盖在 100,081 个聚类中）为某个攻击阶段。这些交易和针对 85 个 Dapp 的攻击有关，对于每个受害 Dapp，研究人员随机抽取 4% 被报告的交易聚类进行人工验证。最终手动统计了 4,003 个交易聚类总计 30,888 次交易。其中 3,671 个聚类与攻击事件有关，3,347 个聚类被正确标记攻击阶段。

表 2: 不同类别的受害 Dapp

类型	Dapps/0-day	攻击者账号/0-day	恶意交易/0-day	知名案例
赌博	51/43	65,778/11,339	360,524/114,473	Lucky Blocks
游戏	28/27	959/919	52,673/52,176	Space War
金融	5/5	183/183	59,872/59,872	STOX
代币	2/1	279/167	4,478/472	Power of Bubble
总计	85/75	67,199/12,608	476,342/226,763	

表 3: 未知集的分析结果

攻击阶段	Dapps/0-day	攻击者 EOA/0-day	恶意交易/0-day
准备阶段	80/70	42,661/8,237	214,408/106,436
利用阶段	85/75	35,955/3,650	143,179/39,908
传播阶段	75/65	18,466/6,545	118,755/80,419

根据表2可知，针对以太坊 Dapps 的攻击很普遍。57.3% 的被害 Dapp 属于赌博类。为了支持赌博功能，这些 Dapp 需要生成随机数，而这些随机数有时是由一个弱的伪随机数发生器完成的，因此受可预测的随机处理漏洞影响。

DEFIER 扫描到的 82% 的 Dapp 都受到了攻击，这可能是因为研究者分析的 Dapp 都是有大量以太币的活跃 Dapp，这使得它们更有可能成为不法分子的目标。另外，在漏洞交易中发现的 85 个受害 Dapp 中，有 75 个（例如 Space War

和 Super Card）以前从未被报告过受到攻击。由此可以看出，DEFIER 在探测未知的恶意合约攻击上是十分有效的。

表3列出了处在不同攻击阶段的 Dapp 的数量。DEFIER 模型发现了与 80 个 Dapp 相关的 214,408 次交易处于攻击准备阶段。在这些交易中，有 507 个功能被攻击者测试，311 个功能在利用阶段被攻击。这表明 DEFIER 模型可以帮助 Dapp 在被利用之前识别出有缺陷的功能。

3 研究意义和展望

对笔者而言，本文相较于其他智能合约的安全研究，另辟蹊径，给预防针对智能合约的攻击提供了新的思路。相较于其他的安全研究着眼点在语言本身或代码编写的安全性上，该研究把预防此类网络犯罪的着力点放到了犯罪生态上。从代码安全拓宽到金融上的安全，以交易序列作为信息输入而非传统安全观念 [5][4] 上的以代码作为输入。

相较于被动的加强代码的安全性，DEFIER 的优势在于能够主动扫描安全威胁，通过使用机器学习的方法从交易信息中判断攻击阶段，在造成损失前阻止攻击者进一步行动。在攻击准备阶段就让 Dapp 开发者意识到存在安全漏洞的函数，及时进行修复。开发者也可以利用 DEFIER 扫描的结果知晓直接攻击者以外的账户，包括资金经理，钱骡和恶意合约开发者等。采用联锁封禁访问的措施，实现及时的风险控制。

同时这篇文章还为后来的研究者拓宽了思路，网络空间的犯罪本质依旧是人进行参与的，底层的漏洞多种多样、防不胜防，但是当我们把安全的视野放到代码以上，从更高层次（如该研究中以交易序列作为信息输入而非合约代码）去分析就能够简化问题（如该研究中对 Dapp 攻击的检测），这是符合当前安全研究趋势的。[1]

DEFIER 依旧是存在一定限制的，它受限于输入的信息：历史交易和交易执行记录。但是他们会错过不产生交易的攻击操作，如本地调用或调用 Dapp 的常量函数，虽然这些操作很少在攻击中被利用，这些问题有待未来的研究者进一步研究。另外作为监督学习模型，DEFIER 需要手工的数据注释，虽然工作量是很大的，但是该数据集的优势在于其目的是捕捉高层次的攻击意图，训练集持续有效直到针对 Dapp 的犯罪生态发生改变。笔者也认为，目前半监督和无监督学习对于分类的效果依旧不佳，使用监督学习模型虽然有样本标注的工作量，但胜在分类准确。

除此以外，笔者认为该研究对于防治未来针对数字人民币的智能合约攻击有着重要意义。该研究为使用循环神经网络定位区块链上的攻击和洗钱活动有

重要参考意义。

希望本文能对您了解智能合约与 Dapp 安全的前沿成果有所帮助。

参考文献

[1] L. Su, X. Shen, X. Du, X. Liao, X. Wang, L. Xing, and B. Liu, "Evil under the sun: Understanding and discovering attacks on ethereum decentralized applications," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 1307–1324.

[2] N. Popper, "A Hacking of More Than $50 Million Dashes Hopes in the World of Virtual Currency," *The New York Times*, Jun. 2016. [Online]. Available: https://www.nytimes.com/2016/06/18/business/dealbook/hacker-may-have-removed-more-than-50-million-from-experimental-cybercurrency-project.html

[3] W. Chen, Z. Zheng, J. Cui, E. C.-H. Ngai, P. Zheng, and Y. Zhou, "Detecting ponzi schemes on ethereum: Towards healthier blockchain technology," in *The Web Conference*, 2018.

[4] N. Atzei, M. Bartoletti, and T. Cimoli, "A survey of attacks on ethereum smart contracts sok," in *Principles of Security and Trust*, 2017.

[5] S. Rouhani and R. Deters, "Security, performance, and applications of smart contracts: A systematic survey," *IEEE Access*, vol. 7, pp. 50 759–50 779, 2019.

[6] S. M. Milajerdi, R. Gjomemo, B. Eshete, R. C. Sekar, and V. N. Venkatakrishnan, "Holmes: Real-time apt detection through correlation of suspicious information flows," *arXiv: Cryptography and Security*, 2018.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[8] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," 2005.

[9] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," *arXiv: Social and Information Networks*, 2016.

[10] "Explore decentralized applications." [Online]. Available: https://www.stateofthedapps.com/