ONCHAIN ANTI-SCAM

YELLOW PAPER

链上防骗黄皮书

骗局分类学·链上取证·分层防御

基于 Onchain Diary 反欺诈研究系列

版本 1.0

2026 年 8 月

Onchain Diary × UZEN Labs

theonchaindiary.com • github.com/UZEN-Labs

目录

摘要

链上骗局已经产业化。批量注册的仿冒域名、模板化的钓鱼合约、自动化的洗钱路径，共同构成一条分工明确的流水线。普通用户面对的不是某个「聪明的骗子」，而是这套流水线。

本手册把 Onchain Diary 反欺诈研究系列的方法收拢成一份系统文档，分三层展开：第一层是**骗局分类学**，把主流链上骗局归为五类（授权类、签名类、合约类、身份类、基础设施类），每一类讲清楚攻击机制，因为防御只有建立在对机制的理解上才靠得住；第二层是**链上取证**，教你用区块浏览器自己读痕迹——查授权、追资金、验合约，这些能力在事前是尽调工具，在事后是自救的第一步；第三层是**分层防御**，按入门用户、活跃 DeFi 用户、机构级三个层级给出可直接执行的检查清单。

本手册坚持一个原则：只写可以验证的事实和可以执行的动作，不复述无法核实的数据。所有方法基于公开链上数据和公开案例。

Abstract

Onchain scams have industrialized. Mass-registered lookalike domains, templated phishing contracts, and automated laundering paths form a division-of-labor pipeline. What an ordinary user faces is not one clever scammer but this pipeline.

This paper condenses the Onchain Diary anti-fraud research series into one systematic document, organized in three layers. First, a **taxonomy** that groups mainstream onchain scams into five classes (approval abuse, signature abuse, contract traps, identity fraud, infrastructure attacks), explaining the mechanism behind each class, because defense only holds when it is built on mechanism. Second, **onchain forensics**: how to read the trail yourself with a block explorer —checking approvals, tracing funds, verifying contracts. Third, **layered defense**: executable checklists for three user levels (beginner, active DeFi user, institutional).

One rule runs through the paper: we state verifiable facts and executable actions, and we do not repeat numbers we cannot verify. All methods rely on public onchain data and public cases.

1 威胁全景

1.1 骗局产业链是一条流水线

链上骗局在十年里完成了从「个体户」到「工厂」的转变。今天的骗局分工明确：有人负责投放（社交媒体假账号、空投诱饵、假客服话术），有人负责收款（批量部署的钓鱼合约和 drainer 基础设施），有人负责洗钱（混币器、跨链桥、分层转移）。用户在任何一环失守，钱就进入整条流水线，几分钟内分散到几十个地址。

这意味着两件事。第一，骗局不会因为你看穿了一个话术就停止——话术是可替换的耗材。第二，防御的重点不是「识别骗子」（骗子的表面身份千变万化），而是守住资产离开你钱包的那几个出口。出口只有两种：**授权（approval）**和**签名（signature）**。这本手册的结构就围绕这两个出口展开。

1.2 为什么传统安全经验不够用

传统互联网的安全直觉在链上会失灵，原因有三：

- **交易不可逆**。银行转账有撤回窗口、有冻结机制，链上交易确认即终局。没有客服，没有申诉入口。
- **授权是持续性的**。一次 `approve` 不是一笔付款，而是一张长期委托书。今天签的授权，明年仍可能被调用。
- **地址无身份**。收款地址可以无限生成，事后追责在工程上近乎不可行。链上取证的产出是证据，不是退款。

这三条决定了链上安全的重心必须前移：把钱骗走之前的五分钟，比之后的一切都重要。

1.3 谁在被骗

被骗的用户画像常被误解为「贪心的人」。实际模式更宽：追空投的新手、用了多年钱包没出过事的老用户、被假客服引导「验证助记词」的普通人、以及被高收益吸引进貔貅盘的投机者。骗局不需要受害者愚蠢，只需要受害者在某个时刻信息不足——第一次领空投、第一次遇到合约报错、第一次接到「官方」私信。本手册第 5 章的案例模式会回到这一点。

2 骗局分类学

分类的意义在于收敛：话术千变万化，机制只有几类。看懂机制，新话术只是旧瓶新酒。

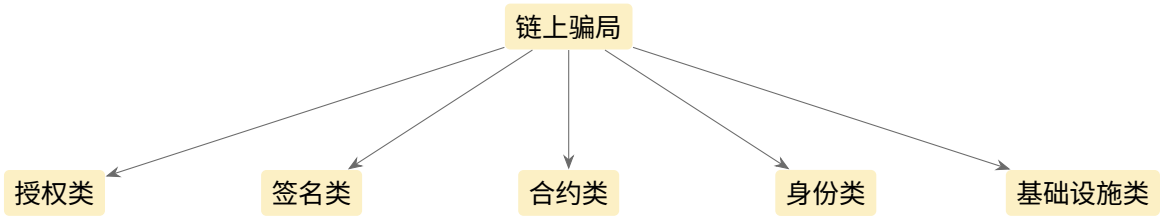

图 1: 五大类骗局。每类针对资产出口的一个环节。

2.1 授权类：一张没写期限的委托书

ERC-20 的 `approve` 授权的是「将来可以从我账户扣款的上限」。问题在于两点：其一，多数 dApp 默认申请**无限授权**（unlimited approval），因为这样可以省掉后续每笔交易的一次 `approve`，用户少签一次、少付一笔 gas；其二，授权不会自动过期。你三个月前在某个交易所签的无限授权，在那个交易所跑路或被黑之后依然有效。

授权类攻击的常见入口：仿冒 dApp 前端诱导你对恶意合约 `approve`；恶意代币空投引导你去「官网」领取，实际领取动作就是授权动作。防御的对应动作很直接：授权前读清楚金额字段（是不是 `uint256.max`），定期用撤销工具清理不用的授权。

Permit 系列把这个问题往前推了一步：`permit` / `permit2` 允许用**签名**代替交易完成授权，不需要 gas、不需要出现在你的地址活动里，恶意站点拿到签名后由攻击者自己付 gas 提交。签名类骗局里会再讲到它。

2.2 签名类：你签的不是你以为的东西

钱包签名界面的翻译层是最大的信息损耗点。`eth_sign`、`personal_sign`、`signTypedData` 在钱包里显示为几行摘要，用户很容易把「登录签名」和「转账授权」看混。签名类骗局的全部技巧就是制造这种混淆：

- **假登录**：站点弹出 `personal_sign`，文案说「登录即表示同意条款」，实际消息体里藏着一笔交易或授权。
- **SetApprovalForAll 钓鱼**：签名内容是 NFT 全量授权。签完，钱包里全部 NFT 可被对方转走。这是 NFT 钓鱼的主力手法。
- **Permit2 钓鱼**：如上节所述，签名即授权，gas 由攻击者出。
- **eth_sign 盲签**：旧接口直接对任意哈希签名，内容完全不展示。正规 dApp 已弃用；见到它，基本可以断定对面有问题。

签名安全的核心原则只有一条：**签名前逐字读显示内容**。找不到解释签名的官方文档，就不签。「稍后再看」不是一个选项，因为签名无法撤销。

2.3 合约类：陷阱写在代码里，但读得出来

蜜罐（honeypot，圈内叫貔貅）：买入正常、卖出被代码拦截。实现方式包括白名单卖出（只有部署者地址能卖）、动态税率（卖出时税率拉到 99%）、余额条件（合约直接改你的账面余额）、时间锁（开盘 N 分钟后禁止卖出）。链上信号很干净：交易记录只有买入没有卖出，合约未开源或开源代码里塞了大量无关逻辑。

Rug pull：三种形态——撤走流动性（部署者持有大部分 LP 且未锁仓）、隐藏铸币权（部署者可随时增发把价格砸穿）、限制卖出（蜜罐的变体）。共同点是权力集中在部署者手里，而链上可以把这三点全部查出来：LP 锁定状态、合约的 mint 函数与 owner 权限、持有人分布。

合约漏洞利用：重入攻击、闪电贷操纵价格预言机、访问控制缺失。这一类受害者通常不是普通用户而是协议本身；普通用户的位置在第四章的「协议尽调」里。

2.4 身份类：绕过钱包，攻击人

身份类骗局不碰合约，碰的是信任。

假空投：社交媒体出现「XX 项目空投查询」链接，域名与官方差一两个字符（`-` 换 `.`、`i` 换 `l`、加 `s`），页面与官方站像素级一致，「领取」按钮弹出的签名是授权或转账。

地址投毒：攻击者生成与你常用地址首尾字符相同的地址，向你的交易历史里塞小额转账。之后你从历史记录里复制地址时，可能复制到投毒地址。零转账变种更隐蔽：攻击者只发一笔 0 金额转账，让投毒地址进入你的记录列表。防御方式朴素但有效：转账前逐位核对首尾各 6 位以上，大额先小额试路。

假客服与恢复骗局：在公开渠道求助（比如在 X 上发帖提到被盗），「客服」私信主动出现，引导你「验证钱包」——把助记词或私钥输入一个「验证网站」。恢复骗局是它的二次收割：被盗后搜索「追回资产」，冒充链上律师或黑客的服务商收取前期费用，然后消失。助记词不存在任何需要输入它的验证场景，这一条足以过滤九成身份类骗局。

2.5 基础设施类：你控制之外的环节

跨链桥攻击：桥是多链架构里资产集中的信任枢纽，历史上多次被盗涉及桥的验证机制缺陷。用户能做的主要是分散：不把资产长期堆在一座桥上，过桥即走。

剪贴板劫持：恶意软件监听剪贴板，发现地址格式就替换。它和地址投毒是同一防御：粘贴后核对。

前端供应链：dApp 前端被入侵或域名过期被接管，即使合约无漏洞，用户也会在前端引导下签恶意内容。应对属于第 4 章的「环境纪律」：重要操作前核对域名，对突然变更的交互流程保持怀疑。

3 链上取证：自己读痕迹

这一章讲三件事：查授权、追资金、验合约。工具是区块浏览器（Etherscan 及各链对应版本），全部免费，不需要 API key。这些技能事前是尽调，事后是自救的第一步。

3.1 查授权：你的委托书清单

在 Etherscan 上打开你的地址页，进入 Token Approvals 页签，能看到你签过的每一笔授权：被授权合约、代币、额度、时间。这份清单就是你的「暴露面」。

读清单的方法：按时间倒序看，凡是想不起来的合约，先查它是什么（下一步讲怎么查合约）；额度为无限的授权，逐个判断是否还需要——不需要就撤销。撤销要付 gas，一次清完比拖沓便宜。

3.2 追资金：钱去哪了

如果已经发生了转账，第一步是把交易哈希存下来（浏览器地址栏里的 `0x...`）。交易详情页给出：资金去向地址、金额、时间、调用方法。资金去向地址点进去看三样东西：持仓（ERC-20 / NFT 页签）、交易对手（它又转给了谁）、地址标签（浏览器社区标注的交易所、混币器入口等）。

洗钱的典型路径是分层：先拆成几十笔小额进不同地址，汇入混币器或跨链，再从另一条链出来。你能做的是把这条路径截图存档——不是因为你追得回来，而是这些证据在报警和联系交易所冻结时用得上。资金如果进了中心化交易所的充值地址，交易所合规部门在接到完整证据链时有冻结先例，这是追回概率最高的一条路径，也是「存证据」的全部意义。

3.3 验合约：三分钟尽调

拿到一个合约地址，按固定顺序看四项：

1. **是否开源**。Contract 页签里源码已验证（绿色对勾）是底线。不开源的合约，签名之前没有任何可说的。
2. **权限在哪**。看 `owner` 能做什么：能不能 `mint`、能不能暂停转账、能不能改税。这些在开源代码里直接读，搜索函数名即可。
3. **持有人分布**。Holders 页签，前几名占多少。部署者 + 关联地址占比过高，砸盘权就在别人手里。
4. **流动性**。如果是交易对合约，LP 是否锁定、锁多久。未锁仓的 LP 随时可以撤走。

这套流程覆盖不了合约的全部风险（代码漏洞需要专业审计），但它能过滤掉大多数明晃晃的陷阱：貔貅、未锁仓、隐藏铸币权，全部在这四项里现形。

3.4 地址标签：地址的身份档案

浏览器和风险评分服务给地址打的标签（交易所热钱包、混币器、drainer 关联、制裁名单）是重要的免费信号。一个「空投资网」项目的领取合约，如果部署者地址的历史与已知 drainer 有关联，这比任何话术都有说服力。地址风险评分把钓鱼关联、制裁曝光、污染传播等信号合成一个分数，机制不神秘，就是上面这些链上痕迹的加权。

4 分层防御

防御不是一堆零散的「注意事项」，是分层结构：底层习惯人人适用，越往上越接近操作纪律。

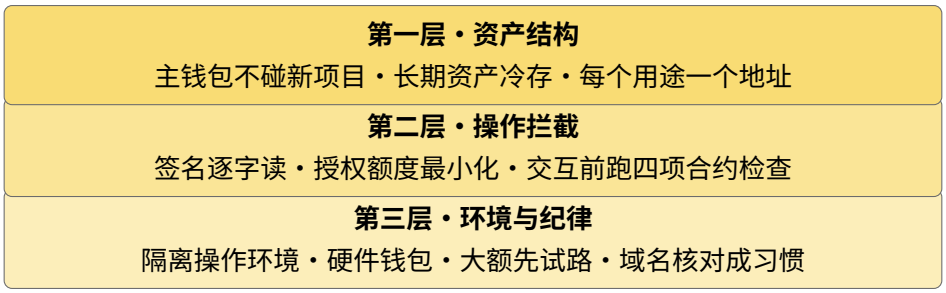

图 2: 防御的三层。下层失效时上层兜底。

4.1 第一层：资产结构

最重要的一个决定在骗局发生之前：资产怎么摆放。

冷热分离。长期不动的资产放冷钱包（硬件设备或离线签名的钱包），日常交互用热钱包。热钱包里的余额是你在任何事故里的最大损失上限——把它设成你能承受的数。

用途隔离。领空投一个地址、交易一个地址、测试新协议一个地址。用途隔离的价值在事故后尤其明显：一个地址被钓鱼，影响范围就是它自己，排查和善后都简单。

4.2 第二层：操作拦截

资产出口只有授权和签名两种，操作层的全部内容就是守住这两个动作。

签名纪律：逐字读显示内容；`eth_sign` 直接拒；分不清「登录」还是「授权」就不签；官方文档找不到对这个签名的解释就不签。

授权纪律：能选限额就选限额，不选无限；授权过的服务不用了就撤销，把撤销当成定期保洁（比如每季度一次）。

合约纪律：新协议交互前跑完第 3 章的四项检查——开源、权限、持仓分布、流动性。四分钟换掉大部分明坑。

4.3 第三层：环境与纪律

域名习惯：从书签进站，不从搜索引擎结果页和私信链接进站；重要操作前核对域名拼写。这一条挡掉假空投和域名仿冒的大部分。

助记词纪律：助记词只存在于纸上或金属板上，不存在于任何联网设备、截图、云笔记、「验证网站」。任何要求输入助记词的场景都是骗局，没有例外。

转账习惯：新地址首次转账先小额试路；粘贴地址后核对首尾各 6 位。挡掉剪贴板劫持和地址投毒。

信息纪律：不在公开渠道暴露自己持有多少资产；求助时注意主动私信的「客服」；对「限时」「错过即无」的紧迫感保持警觉——紧迫感是话术的标配，不是机会的属性。

4.4 机构级补充

管理他人资产的团队（DAO 金库、基金、协议团队）在第 1–3 层之上还需要：多签作为资产出口的强制环节（单签权限最小化）；操作流程成文（谁提案、谁批准、时间锁多久）；对合约交互白名单化管理（新合约上线前完成审计或多重独立审查）。历史上多起机构级损失的原因不是没有多签，而是流程被绕过——制度的价值在它被执行的时候才存在。

5 案例模式

本章不点名具体案件，也不引用无法核实的损失数字。我们只归纳模式——模式是可复用的，数字不是。

5.1 模式一：空投查询链

一位用户在社交媒体看到「老用户空投查询」链接。域名与知名项目只差一个字符。页面是官方站的完整复刻。连接钱包、「查询资格」弹出 `personal_sign`，用户以为是登录。实际上签的是 Permit2 授权。攻击者自付 gas 提交，用户钱包没有任何可见动静。数日后代币被划走。

这个模式里用户的三次机会：入口（从书签进官方站）、签名（逐字读——Permit2 授权在钱包里显示为 allow spend 的内容）、授权面（定期检查 Token Approvals，发现想不起来的授权就撤销）。

5.2 模式二：假客服二次收割

一位用户钱包被盗，在社交媒体发帖求助。一小时内三个「安全团队」私信出现，都声称能追回。其中一个引导他到「链上验证门户」输入助记词以「生成追踪报告」。他照做，第二个钱包也被清空。

教训是一条硬规则：助记词没有合法输入场景。主动出现的帮助者，先按骗子处理。

5.3 模式三：貔貅盘的完整周期

一个 meme 币上线，社群热度高、涨幅陡。早期买入者确实能卖出（这是饵）。新进者涌入后，卖出开始失败——合约里的税率函数被激活。部署者地址此时撤走未锁仓的 LP。链上信号从第一天就存在：合约代码里有按部署者开关的税率逻辑、LP 未锁定、持仓前几名是关联地址。只是没有足够多的人在买入前花四分钟跑一遍第 3 章的检查。

5.4 模式四：地址投毒得手

一位用户习惯从历史记录复制收款地址。攻击者此前向他的历史里塞了一笔小额转账，投毒地址与常用地址首尾各四位相同。某次大额转账，他从列表里复制了错的那个。交易确认即终局。

核对首尾各 6 位、大额先试路，这两条习惯每一条都能单独拦住这次事故。

5.5 模式之间的共性

四个模式对应四个不同的入口（空投、客服、代币、转账），但防住它们的动作高度重叠：书签进站、逐字读签名、四项合约检查、核对地址、助记词零输入。这就是第 2 章分类学的意义——话术在变，出口不变，防守的焦点也不变。

6 附录 A：核心术语

以下术语精选自 Onchain Diary 术语库（完整版 220 条见 `https://theonchaindiary.com/glossary/`），覆盖本手册全部概念。

授权（Approval）
ERC-20 标准中允许被授权合约在限额内扣用你代币的操作。授权是长期有效的，不会自动过期。

无限授权（Unlimited Approval）
授权额度设为最大值的做法。省 gas，但把扣款上限完全交给对方合约。

Permit / Permit2
用签名代替交易完成授权的机制。对用户的好处是免 gas；被滥用时的效果是「签名即授权」，且授权动作不出现在你的地址交易列表里。

盲签（Blind Signing）
钱包对无法解码的内容签名。`eth_sign` 是典型。正规 dApp 已弃用，见到即红旗。

SetApprovalForAll
NFT 全量授权。一次签名交出整个系列的转移权，是 NFT 钓鱼的主力手法。

钱包排水器（Wallet Drainer）

一类钓鱼即服务（phishing-as-a-service）工具，向骗子提供签约页面、签名诱导和资产转移的整套基础设施，按窃取金额抽成。

蜜罐 / 貔貅（Honeypot）

买入正常、卖出被合约逻辑拦截的代币陷阱。常见实现：卖出白名单、动态税率、余额篡改、时间锁。

Rug Pull

项目方卷款离场的统称：撤流动性、增发砸盘、限制卖出，或组合使用。

LP 锁定（Liquidity Lock）

流动性代币被时间锁合约托管。LP 未锁定的交易对，流动性随时可被部署者撤走。

地址投毒（Address Poisoning）

向目标地址的交易历史里塞入首尾字符相同的仿冒地址，诱导复制粘贴出错。零转账变种只留地址记录不动资金。

混币器（Mixer）

打断资金与来源关联的服务。攻击者洗钱路径的常用环节，多受制裁监管关注。

重入攻击（Reentrancy Attack）

合约在外部调用未完成时被重复调用的漏洞。历史多起协议被盗的直接原因。

闪电贷（Flash Loan）

同一区块内借还的无抵押贷款。本身是中性的 DeFi 原语，被用于攻击时充当免本金的操纵资金。

三明治攻击（Sandwich Attack）

MEV 的一种：机器人把你的交易夹在两笔交易之间，通过你造成的价格偏移获利。

助记词（Seed Phrase）

钱包的根凭证。任何要求输入助记词的网站和服务都是骗局，没有例外。

多签（Multisig）

多把私钥共同控制的地址。机构资产出口的标准配置。

时间锁（Timelock）

关键操作需等待固定时间才生效的合约机制。给社区留出反应窗口的治理标配。

地址标签（Address Tag）

区块浏览器或评分服务对地址的身份标注（交易所、混币器、drainer 关联等），免费且高频更新的信号源。

7 附录 B：快查清单

撕下来贴墙版。每一项对应正文章节。

每次签名前

- □ 逐字读完钱包显示的签名内容
- □ 说不清这是什么签名 → 不签（§2.2）
- □ `eth_sign` → 直接拒（§2.2）

每次授权前

- □ 额度能选限额就不选无限（§2.1）
- □ Permit/Permit2 签名按授权对待，不按登录对待（§2.1）

新代币 / 新协议交互前（四项检查，约四分钟）

- □ 合约开源且已验证（§3.3）
- □ 读 `owner` 权限：mint / 暂停 / 改税（§3.3）
- □ 持有人分布：前几名占比（§3.3）
- □ 流动性锁定状态与期限（§3.3）

每次转账前

- □ 粘贴后核对地址首尾各 6 位（§2.4）
- □ 新地址 / 大额 → 先小额试路（§4.3）

每季度一次

- □ 清理 Token Approvals：想不起来的授权全部撤销（§3.1）
- □ 检查资产结构：热钱包余额是否仍在你承受范围内（§4.1）

永远

- □ 助记词只写在纸上，零联网输入（§4.3）
- □ 从书签进站，不从搜索结果和私信链接进站（§4.3）
- □ 主动私信的「客服」按骗子处理（§5.2）

参考来源

本手册的方法与机制描述提炼自 Onchain Diary 反欺诈研究系列（https://theonchaindiary.com/），主要包括：钓鱼攻击全景、签名骗局五种变体、无限授权风险、蜜罐识别六信号、Rug Pull 三形态、地址投毒、假空投产业链、链上分析入门、被盗资产追踪等 76 篇文章与 220 条术语库。LaTeX 源码与历史版本：https://github.com/UZEN-Labs/antifraud-yellowpaper。

链上防骗黄皮书 v1.0·Onchain Diary × UZEN Labs ·2026 年 8 月