智能合约安全之
形式化验证研究报告

2018.06.27

引言

随着平台级应用的普遍化，智能合约涉及的金额呈指数级别增长，智能合约的安全问题也成为投资者和开发者共同关注的焦点。今年以来有数个基于 ERC-20 的 ICO 项目因为智能合约代码出现漏洞而遭到黑客攻击，导致投资者巨额的损失。为了防止类似事件的发生，交易所、钱包、项目方等都在智能合约安全上加大投入，同时围绕着智能合约安全的周边生态成为目前投资的热点。

作者

ONE.TOP 洪妙丛

节点研究中心 蔡晨曦 武怡 郎瀚威

支持机构（排名不分先后）

金色财经、BlockMasterMail、babi 财经、金塔行情、星球日报、节点财经

（本报告由 ONE.TOP X 节点研究中心联合发布）

目录

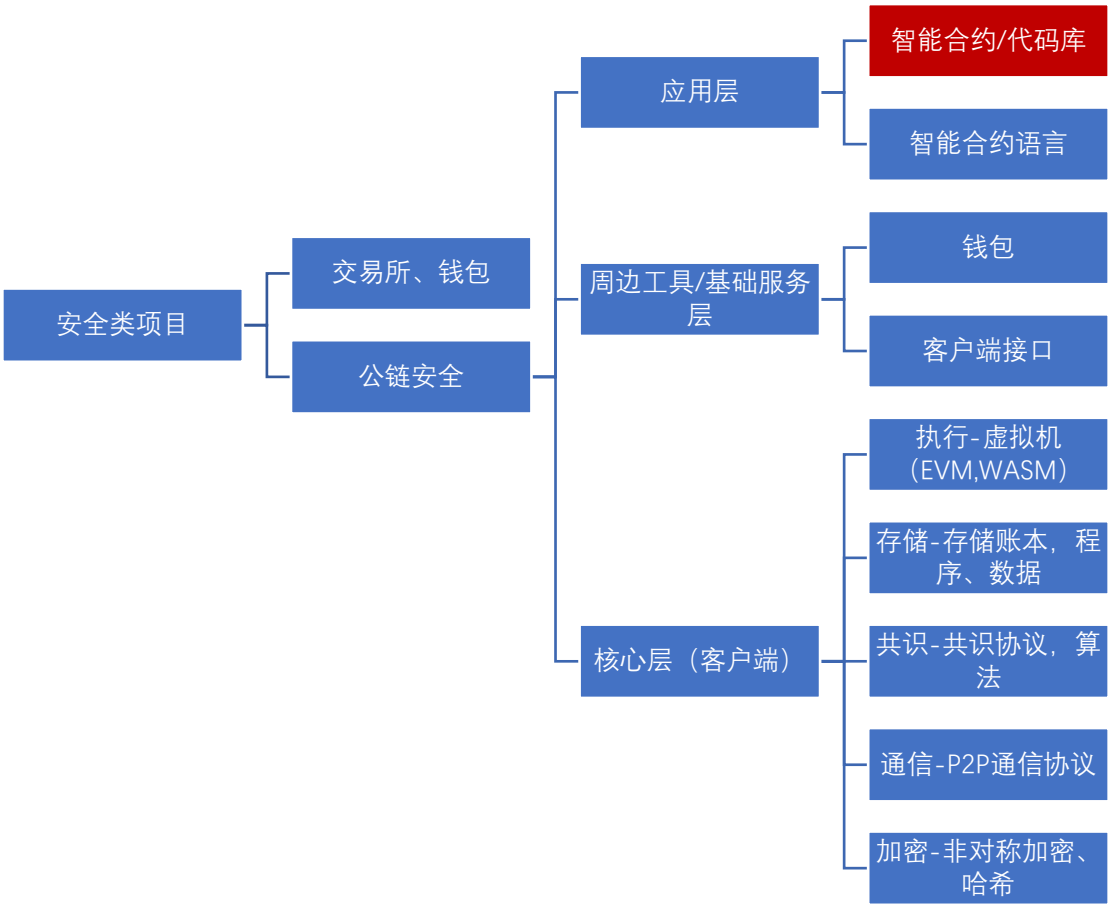
1 区块链安全领域分类——智能合约安全的分类

智能合约安全方面的措施总体来说分为以下几个大类，合约开发模板、合约审计、智能合约语言设计和赏金猎人机制。

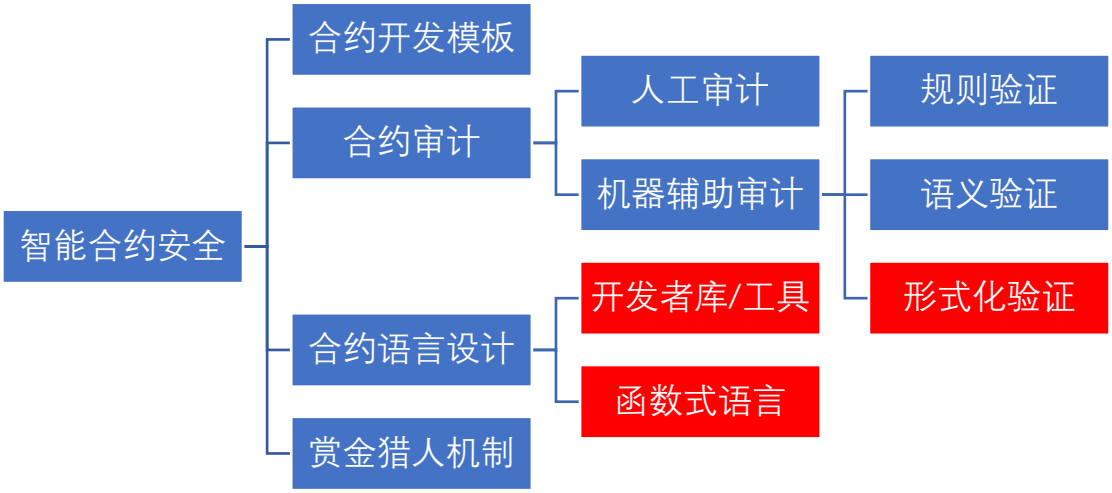
1）合约开发模板如 OpenZeppelin、Etherparty 等为标准化的合约提供经过多次实战验证的标准化模板，在节约时间同时保证合约安全性。

2）合约审计的方法又分为人工审计和机器辅助审计

3）机器辅助审计又分为规则验证、语义验证和形式化验证，而形式化验证是本文关注的重点。



引用自 consensys 唐奕 《公链安全》



智能合约语言设计中，许多公链平台开始采用 Haskell 或 OCaml 一类的函数式语言，因为函数式语言更为便于编写形式化验证方面的开发者库和工具。赏金猎人机制，相对于前几项措施，更多是部署合约后发现漏洞的弥补机制。

形式化验证指的是用数学中的形式化方法对算法的性质进行证明或证伪。方法有两种：

一种是模型检验，即把系统所有可能的状态列出并进行一一检验，此种方法全自动化但只适合小型系统；

另一种是演绎验证，首先把系统代码标记成抽象数学模型，然后对定理进行证明，此种方法适合大型系统，但是需要首先人工将系统的运作方法转换成验证系统可以理解的语言。

2 为什么关注形式化验证

目前为止，形式化验证主要应用在军工、航天等对系统安全非常高的领域，在消费级软件领域几乎没有应用。由于传统互联网应用与区块链应用的运行环境有着本质的不同，其开发流程也应当相应地进行调整，其中最关键点在于安全验证环节的投入比例。

在传统互联网应用中，由于普遍采用中心化服务器+客户端的模型，如果应用出现安全隐患只需要对服务器端代码进行修改就可以轻松排雷，并且服务器端可以对用户数据进行回滚以挽回用户损失。因此，传统互联网应用开发的过程较为注重快速迭代，以牺牲安全性换取效率和功能上的快速升级。

在区块链应用中，由于区块链的不可篡改性，智能合约一旦上线并出现安全隐患，对用户造成的损失是巨大且不可挽回的。一旦出现黑客事件，需要整个社区的共识才能回滚交易，所以每次遭受攻击都回滚交易也是不现实的。因此，区块链应用开发的过程需要用大量的测试和检验以获取足够的安全性，而反过来牺牲迭代的速度。

由于区块链开发人员的稀缺，远远无法赶上智能合约数量的增长，人工审计智能合约是成本非常高昂的，因此机器辅助验证是目前的必然趋势。规则、语义验证的实现，相对较为容易，技术门槛也较低，但是只能进行一些浅层的纠错，不能深入程式的逻辑。因此，只有形式化验证方法有希望真正提高智能合约审计的自动化程度。

3 目前市场上的形式化验证相关产品

目前区块链产业中与形式化验证相关的产品可以分为三类: Vaas 平台，公链，和语言。

项目名称	分类	描述	创始团队背景
CertiK	Vaas	结合证明引擎和赏金猎人的综合性安全验证平台	哥大/耶鲁
Imandra	语言	OCaml 子语言，专注于金融交易系统的形式化验证	伦敦金融城/英国剑桥
链安科技	Vaas	提供多个区块链平台验证工具，以及将合约代码转成定理的证明工具	电子科技大学
The Matrix	公链	基于 AI 辅助的形式验证以及动态约束检查	清华/北大
Securify.ch	Vaas	一键对以太坊智能合约进行形式化验证	ETH Zurich
Runtime Verification	Vaas	使用自己开发的 K 框架对虚拟机二进制码进行形式化验证	UIUC
Tezos	语言	采用函数式编程语言 Michelson 作为智能合约语言，方便形式化验证	华尔街/R3

3.1 Vaas 平台

Vaas 平台是直接面向开发者提供形式化验证服务的平台。目前 Vaas 类项目包括 CertiKzecurify.ch、Runtime Verification 等项目。目前，CertiK 仍在

募资阶段，链安科技据称已经有研发成功及获得专利的产品，Securify.ch 的测试版已经上线，而 Runtime Verification 已经在商业运营。

与其它几个项目不同，Runtime Verification 是基于 EVM 虚拟机二进制码进行形式化验证，而非针对智能合约本身用的高级语言，因此在安全性上又更进一步，避免了因编译器编译过程中可能产生的漏洞。

3.2 语言

语言类产品一般为函数式语言的子语言，提供与智能合约形式化验证相关的开发者库和工具，目前有 Imandra 和 Tezos 等项目。

其中，Imandra 发布了一套开源的以太坊虚拟机用 ImandraML 语言标记的模型，并且专注于交易所等金融应用场景的形式化验证，用以确保金融交易的合法合规，据称相关技术已经用于华尔街顶级投行的交易系统。

3.3 公链

直接包含形式化验证引擎的公链产品目前只有 The Matrix 项目，特征是基于 AI 辅助的形式化验证及动态约束的检查。AI 是否对于形式化验证的自动化带来帮助在技术上仍是个未知数，因此这个项目也将成为这个领域的试金石。

4 详细对比

	Certik	链安科技	runtime verification
创始人	邵忠，耶鲁大学计算机科学博士；顾荣辉，哥伦比亚助理教授	杨霞女士，电子科技大学副教授、博士后	Daejun Park，首尔国立大学计算机科学与工程学士学位和硕士学位，目前正在攻读博士学位
核心团队	核心成员来自哥伦比亚大学、耶鲁大学和普林斯顿大学，专业背景都是计算机技术，团队技术实力强硬	20 多名来自海外知名高校和实验室(CSDS/耶鲁/UCLA)留学经历的副教授、博士后、博士、硕士组成	核心团队成员大部分都有形式化验证方向的研究和学习经历
合作伙伴	量子链、INT、菩提	火币网、OK 资本、比原链、布比区块链、云象区块链	NSF、NASA、Ethereum、Boeing
针对市场	对基于以太坊上开发的 DAPP 和系统进行形式化验证	对智能合约进行形式化验证，支持以太坊和 EOS	致力于飞机，航天器和区块链领域的软件系统的安全性，可靠性和正确性
主要技术	开发了一个基于形式化验证的平台，创新的使用了包括智能标签框架、层级分解在内的技术，帮助实现自动化的形式化验证	高度自动化的智能合约安全审计平台 VaaS，再以人工方式对智能合约代码逐行复核，保证审计质量	用自己研发的 runtime 验证技术对对智能合约进行形式化验证工作

5 产品简介

5.1 Certik

Certik 是一个是形式化验证框架，经过 certik 验证的智能合同、DApp 以及区块链将会被附上证书形式的标志，来展示其正确性和安全性。Certik 平台提供的主要功能包括以下部分。

智能标签框架

CertiK 平台应用深度学习技术来构建智能标签框架，有了这个框架，大多数共享逻辑和属性代码（前置条件，后置条件，不变量等）都可以被自动标记，从而使得程序的逻辑，语义更加清晰和规范，这样验证的工作量就可以大大减少。

基于层的分解

这种技术揭示了分层设计模式的见解，并使得将复杂的证明任务分解为更小的任务成为可能，并在适当的抽象层面验证它们中的每一个。

可插拔的验证引擎

提供开放式的协议，更先进算法的证明引擎可以自由插入这个系统。

机器可检验的证明对象

构建机械式的证明对象（或者反例），这些证明对象可以快速的被任何人检验，同时作为验证程序的证书（机械式证明对象可直接作为证书，赏金猎人提供的证明对象，需要检察官进行人工检验后才能作为证书。注：赏金猎人和检察官后文会进行描述。）

认证的 DAPP 库

为集成开发环境提供了一系列认证库和插件，以便开发出质量更高的 dapp，但是会花费一些 CTK。

定制化的认证服务

如果有高可靠性要求，可以提供定制化的认证服务，由验证领域的专家提供帮助并出具综合报告。

Certik 平台围绕 bug 的检测和修复建立了一个去中心化的生态，该生态由五个角色构成，分别是客户、赏金猎人、检查官、社区贡献者以及开发使用者，该生态工运行模式如下图所示。

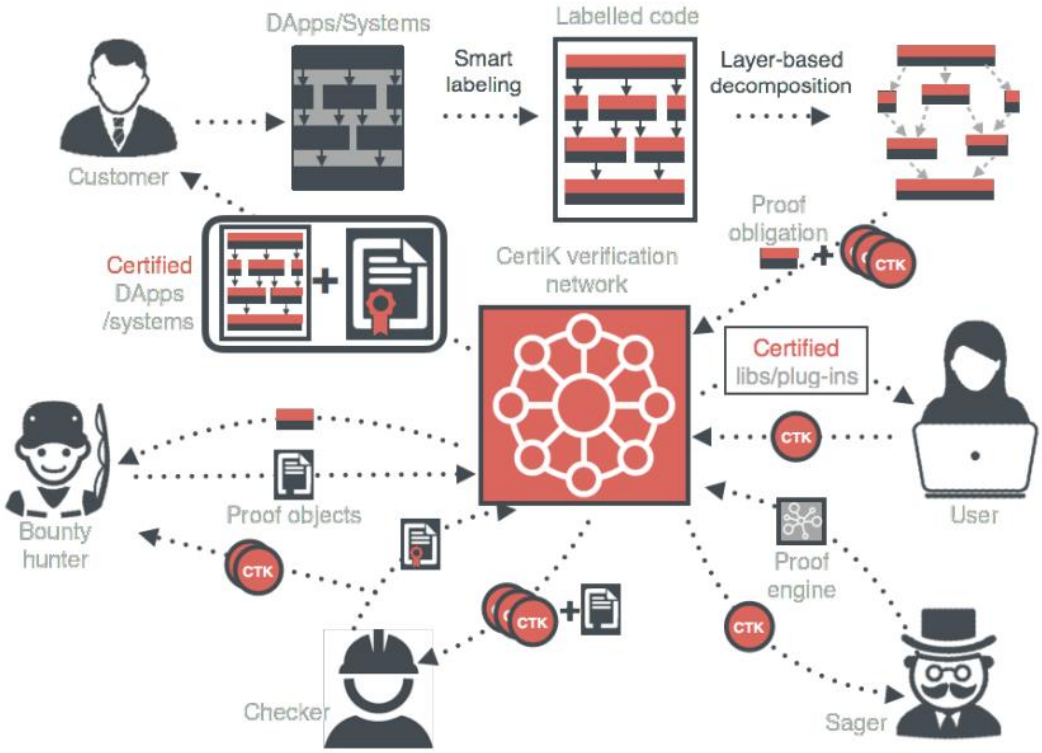
该系统可以分为机械化部分和人工部分。客户在 certik 平台上提交 dapp 或者程序系统，平台自动为其添加智能标签，并自动进行分解，形成小模块的证明任务，这个环节客户需要消耗一定量的 CTK。

分解完小模块后，系统由两种方式进行验证，简单的证明任务可以由一些自动验证器(例如 SMT 解算器)解决。除了平台内部自带的验证器（证明引擎），CertiK 平台提供开放协议，社区贡献者可以将带有更高级的求解算法的证明引擎自由地插入到该系统中，并获得一些 CTK 奖励。赏金猎人可以随机使用他们的引擎，并进行评估，优秀的引擎将被社区研究和推广。

另一种验证方式针对较为复杂的证明任务。赏金猎人接到该任务后，构建一个证明对象并进行广播，接着检查官对证明对象进行检测，当证明对象验证通过后，会被贴上证书的标签，赏金猎人和检查官分别获得 CTK 奖励。

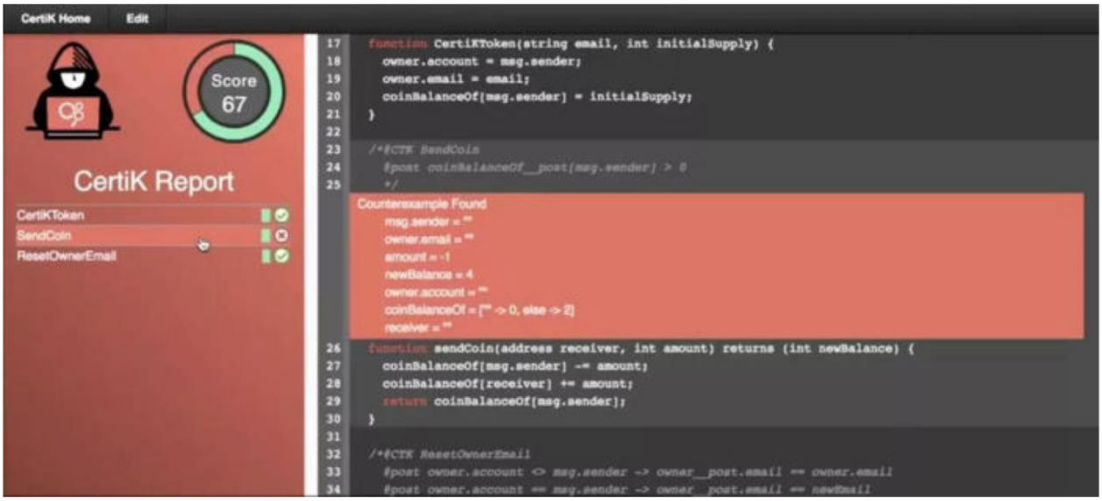
所有分解的证明任务被验证后，CertiK 平台的后端将返回详细而全面的评估报告。

开发使用者可以使用所有 CertiK 平台的认证库和 IDE 插件， 构建自己的 DApp /系统，这需要花费一定的 CTK。

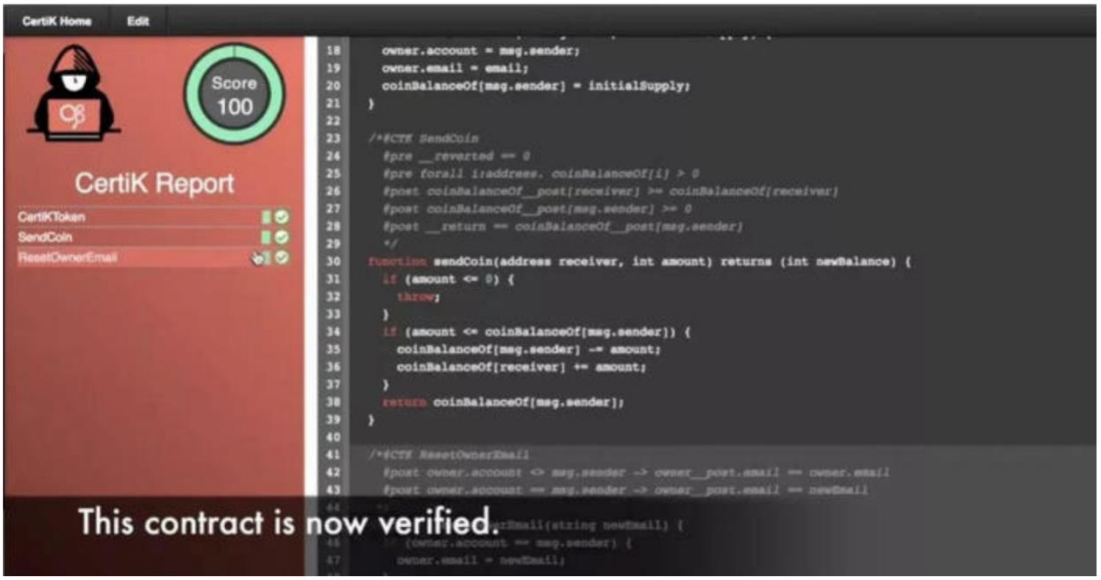


以下是 certik 的操作界面

打开 CertiK 的系统，上传要检测的智能合约，按下检测按钮。



检测完毕后， CertiK 会提供这份智能合约的安全系数，并告诉你哪一块程序存在着错误，并提供详细的解决方案。



5.2 链安科技

提供几种验证服务：

- 第一个，安全审计
- 第二个，开发、审计一条龙
- 第三个，合约开发

其中安全审计模块针对的漏洞包括代码编程规范漏洞、代码逻辑漏洞、函数调用漏洞、整型溢出漏洞、可重入攻击漏洞、执行顺序依赖漏洞、时间戳依赖漏洞、平台接口误用漏洞。

链安 CEO 通过一个例子从数学原理上对形式化验证进行了描述说明。以近期频发的溢出类安全漏洞属性检查为例，如检查代码

int8 c=a+b

是否存在溢出漏洞，下面展示对这行代码的功能正确性和安全属性的证明过程。

首先，对整数类型建模，定义形式化规则：

Int8.repr: Z -> int8

该规则通过截取纯数学整数（取值范围从无穷小到无穷大）的低 8 位数值得到一个 8 位长度的机器整数。然后写加法运算的形式化规范，如下：

{a:int8,b:int8} //

设置代码执行的前提条件，保证 a 和 b 的类型是 8 位有符号机器整数;

{c = a + b;} //

加法运算的源码程序；

{(int8.repr(a+b)) /\ ((Int8.repr (a+b)) = (a+b))} ; //

设置代码正确执行的后置条件。

其中，*(int8.repr(a+b)*描述，

是为了证明代码的功能正确性是否满足，即需要证明源代码是对 a 和 b 进行求和而不是求差或任何其他运算逻辑，并且将运算结果转换为 *int8* 类型。此外，需要对是否溢出的安全属性进行证明，因此添加后置条件：

((Int8.repr (a+b)) = (a+b))

因为一旦

a+b>int8.max_singed 或

a+b<int8.min_signed 都将导致

(Int8.repr (a+b)) ≠ (a+b)。

最后，根据前置条件证明代码的执行是否满足上述后置条件。如果产生一个不可证明结果，说明程序功能不正确，或者存在溢出安全漏洞。然后根据证明结果，对源程序进行分析修改，然后再重新证明，直到证明通过为止。链安采用这种数学的证明方式将代码形式化描述为公式，并对其进行逻辑漏洞和安全漏洞证明，基于此原理，实现了自动化的验证工具，能够方便、快速地验证出代码的功能正确性和安全属性。

5.3 runtime verification

Runtime Verification Inc.是一家初创公司，通过使用自己研发的 runtime 验证技术致力于提高汽车，飞机，航天器和区块链领域的软件系统的安全性，可靠性和正确性。从 2001 年开始，runtime verification 就成为 NASA 的研究科学家。

Runtime 验证技术依赖于程序执行不应违反某些属性的原理。 其中一些属性，如并发程序中缺少数据竞争，具有通用性，可以自动检查。 其他属性，如专用库的规格，是针对特定应用程序或用途定制的。Runtime 验证技术可以自动检查通用属性，不需要开发输入，并且可以检查开发人员用形式化方式表达的任何定制属性。

在区块链方面，runtime 公司主要对智能合约进行形式化验证工作，验证步骤包括：

（1）形式化

根据智能合约所有者提供的非形式化规则，对智能合约的高级业务逻辑进行

形式化。

（2）确认

形式化后的高级业务逻辑需要由智能合约的所有者确认，可能需要经过多轮讨论和修改，才能确保它能正确捕捉合同的预期行为。

（3）提炼

通过多个步骤将形式化后的规则完善到虚拟机级别，以捕获虚拟机特定的细节。 最终的虚拟机级别规则的作用是确保在字节码级别没有意外发生，也就是说，只有在执行字节码时才会发生高级规则中指定的内容。

（4）编译和测试

使用与部署合同相同的编译器版本，将智能合约从其高级代码编译为生成的虚拟机低级代码。

（5）验证

最后，对智能合约的虚拟机字节码与虚拟机的规则进行形式化验证，通过这样的方式不用依赖编译器的正确性。同时，runtime 使用自己研发的 K-framework 结构演绎验证程序， 以达到严格推理虚拟机字节码而不遗漏任何虚拟机怪癖的效果。

6 智能合约的重大漏洞例举

6.1 TheDAO

2016 年 6 月 17 日，一名黑客在编码上发现了漏洞，使得他可以从 The Dao 上抽走资金。在攻击的头几个小时，360 万的以太被转出，在当时价值相当于七千万美元，今天则达到了 21 亿美元。黑客达成了他想要的破坏，退出了攻击。

在此漏洞中，攻击者能够“请求”智能合同(DAO)多次返回以太，且都是在智能合约更新它的余额前进行的。两个主要问题使它成为可能：一是在创建 DAO 智能合约时，编码人员没有考虑到递归调用的可能性；二是智能合约首先发送 ETH 资金，然后再更新内部 token 余额。

重要的是要了解这个 bug 不是来自以太坊本身，而是来自基于以太坊上的构建应用程序。为 DAO 编写的代码有多个缺陷，递归调用的漏洞就是其中之一。

另外一种理解它的方式是比较。以太坊比作是互联网，基于以太坊的应用比作是网站。也就是说，如果网站不运行，不意味着整个互联网出问题。它只能说明网站有问题。

黑客出于未知的原因停止从 The DAO 抽取资金，尽管他可以继续这么做。以太坊社区和团队很快就控制了局面，并提出了多项应对攻击的建议。

然而，这些资金被存入一个账户，有一个 28 天锁定期，黑客无法转走。为了退还损失的钱，以太坊通过硬分叉把被黑资金退还到原所有者的账户上。退还汇率是 1 ETH 兑 100 DAO ，与首次公开发行时的汇率相同。

毫无意外，黑客攻击意味着 DAO 的终结。很多以太坊用户质疑硬分叉违反区块链的基本信条。更糟糕的是，2016 年 9 月 5 日，Poloniex 交易所下架了 DAO token，Kraken 在 2016 年 12 月也下架了 DAO token。

与美国证券交易委员会(SEC)2017 年 7 月 25 日发布的报告相比，以上的这些问题都相形见绌。它提到：

“由一个名为“DAO”的“虚拟”组织提供和出售的代币是证券，因此受联邦证券法的约束。报告确认，发行基于区块链技术的证券发行者必须登记此类证券的发行和销售，除非有有效的豁免。参与未注册发行的证券的人也可能要对违反证券法的行为负责。”

换句话说，TheDAO 的发行与首次公开发行(IPO)一样，受到同样监管原则的约束。按照 SEC 观点，DAO 违反了联邦证券法，所有投资者也一样。

6.2 EDU

区块链项目 EDU 智能合约中存在漏洞。在一个名为 transferFrom 函数中，缺少 Safemath 验证，利用溢出攻击可以让攻击者从任何一个 EDU 余额不为 0 的账号内向另外一个账号转出 任意数量的 EDU Token。

由于目前 EDU 在火币 pro 上线，黑客利用漏洞累积向火币转入了超过 20 亿枚 EDU。这也导致了 EDU 的价格崩盘。

7 其他常见安全类项目介绍

7.1 Sentinel Protocol

韩国团队，基于 ICON 的第二个项目。项目目的是要创建一个去中心化的声誉系统，通过集体智能和人工智能相结合，将所有数字货币的粘片，黑客信息，可疑钱包地址等各种信息记录在区块链上。

5.2 Atonomi

要成为世界上第一个分布式物联网安全协议。

Atonomi 利用区块链上的特性与经济激励，利用 Token 注册和验证评估系统，构建出一个物联网设备的认证与信誉数据库，保证每一台加入网络的设备，都拥有良好的信誉。同时，应用母公司 Centri 的技术（Centri 是传统互联网的安全公司，拥有多项安全方面的技术专利，并与很多互联网大公司是合作伙伴）给设备加密，确保数据隐私并阻止它被黑客利用来连接到别的物联网设备。同时 Atonomi 的 Token 也像 IOTA 一样支持设备之间的微支付需要。

5.3 Gladius

严格意义来讲，Gladius 并不是一个保护区块链的项目，而是一个利用区块链技术，来保护传统互联网的项目。

Gladius 是利用存储和流量的经济激励，来创建基于区块链的分布式 CDN，从而实现 DDoS 攻击的分散保护，同时创建一个大型流量池，处理不断出现的 DDOS 请求，提供比传统 DDOS 防护更加经济和高效的方式。

Gladius 的桌面客户端允许用户租用计算机空余的带宽，并为提供者奖励代币，代币可以购买区块链服务，也可以用于开发，有点变向挖矿的意思。

5.4 Quantstamp

正如一家公司的财务报表需要审计一样，智能合约，同样需要审计，有了审计，这些类似漏洞发生的概率，可以被大幅度降低。

简单说来，Quantstamp 就是这样一个区块链智能合约的审计类项目，让智能合约变得更加安全。当然了，和像四大会计事务这样的中心化组织不一样，咱们是去中心化的。

5.5 POLYSWARM

Polyswarm 第一个去中心化的杀毒软件市场

去中心化正是区块链的看家本领，Polyswarm 是世界上第一个尝试用区块链来改变这个市场的。他们设计的系统里有 4 种角色。

用户（End Users）报告可能染毒的文件信息，并在平台上悬赏，得到及时而准确的安全信息。

安全专家（Experts）通过分析病毒信息来决定这是否是一个病毒来赚取奖

励。

8 形式化验证投资逻辑的展望

由于智能合约形式化验证目前尚处于起步阶段，目前项目大多还没有落地。展望相关行业需求，相关的投资逻辑如下。

8.1 金融领域细分首先落地

目前为止，造成严重后果的漏洞大多来自钱包、ICO 募资等金融领域应用的合约，而这类合约逻辑较为标准化，相对较容易用形式化验证的方式进行查验。因此，专注于金融领域的形式化验证产品更有潜力。

8.2 函数式语言长期来看是趋势

许多新的公链项目，如 Aeternity、Tezos 等都采取了函数式语言作为智能合约的编程语言，目的在于方便形式化验证工具的开发。因此，在选择公链项目上，函数式语言是首选。

9 节点研究中心介绍

关于节点研究中心


创始合伙人
杜均


管理合伙人
杨玉梅


研究中心负责人
郎瀚威

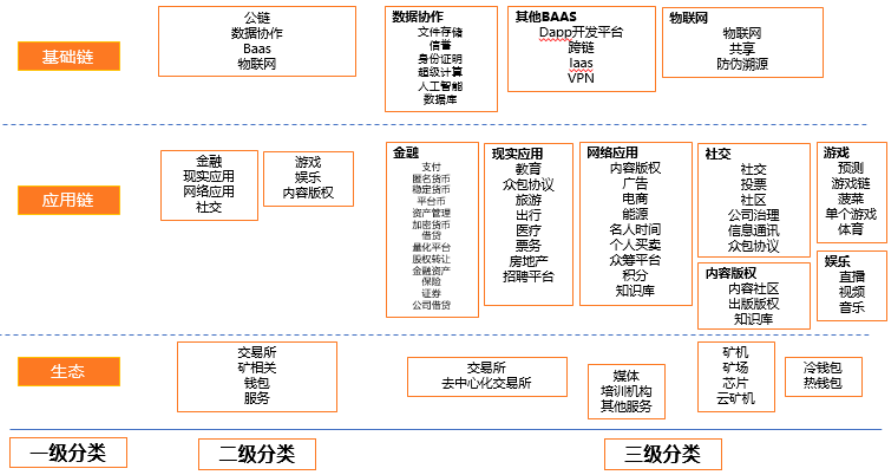

分析师
朱子川


分析师
武怡


助理分析师
蔡晨曦

- 节点研究中心 专注于区块链行业分析与投资
- 团队氛围良好 高效友善开放活泼 分析组成员来自清北复及海外高校
- 简历投递:
  langhanwei@nodecap.com

节点资本项目分类



如征得本公司同意进行引用、刊发的，须在本公司允许的范围内使用，并注明本报告的发布人和发布日期，提示使用本报告的风险。