区块链安全研究中心

Blockchain Security Research Center

区块链智能合约安全审计白皮书（2018年）

上海交通大学 网络空间安全学院

CAICT 泰尔终端实验室

中国信息通信研究院 China Telecommunication Technology Labs-Terminals

掌御科技

检测单位：

上海交通大学网络空间安全学院

中国信息通信研究院泰尔终端实验室

上海掌御信息科技有限公司

联合发布单位：

中国区块链应用研究中心

杭州加密谷区块链科技有限公司

上海淳粹文化传媒有限公司

版权申明

免责申明

本《区块链智能合约审计安全白皮书（2018 年）》（简称《白皮书》）由区块链安全研究中心（由上海交通大学网络空间安全学院、中国信息通信研究院泰尔终端实验室、上海掌御信息科技有限公司联合成立）采用公开、合法的信息，运用相应的科学研究方法，对区块链智能合约安全检测平台（www.sjtubsrc.net）内 31276 份智能合约进行检测，得出智能安全合约在九大安全漏洞的类型分布并对每种类型的漏洞严重等级进行评级。本次检测不对企业数据、用户隐私造成任何破坏，旨在发现区块链智能合约审计安全本身的问题，促进区块链行业发展，对于存在的安全问题不做深入利用。

本《白皮书》的检测以区块链智能合约安全检测平台（www.sjtubsrc.net）内 31276 份智能合约为数据样本并进行同等测试、科学统计、客观评定，过程无任何主观因素及人为干预；

本《白皮书》代表参与单位区块链安全研究中心（上海交通大学网络空间安全学院、中国信息通信研究院泰尔终端实验室、上海掌御信息科技有限公司）观点，旨在对智能合约的安全漏洞进行类型的分布统计分析及对漏洞严重等级进行评级，加强区块链行业从事着对智能合约安全漏洞的了解，引起相关人群对智能合约安全的重视，促进区块链行业发展，仅供读者参考。相关机构或者用户须根据情况自行判断。测试单位力求在白皮书中提供信息的完整和准确性，但是并不能保证信息的完整性和准确性。白皮书中提供的数据、观点、文字等信息不构成任何法律证据，不代表官方机构意见。如果白皮书中的研究对象发生变化，测试单位将不另行通知。对白皮书数据有异议，可以联系区块链安全研究中心（微信公众号：sjtubsrc）。

未获得测试单位的书面授权，任何人不得对本白皮书进行任何形式的进行有悖原意的删节和修改。如引用、刊发，需联系区块链安全研究中心（微信公众号：sjtubsrc）进行授权并注明来源。

本《白皮书》是测试单位根据公开数据，依据专业模型和算法，进行公平、严谨测试、计算和分析得出的研究成果，不收取任何费用，不涉及到任何商业行为。

如有任何疑问，请联系区块链安全研究中心（微信公众号：sjtubsrc）。

前言

2018 上半年，以太坊全球日均新增合约地址数量超过 3500 个，日均新增个人地址数量超过 85000 个。受数字货币行情的影响，一季度新增数量明显高于二季度，二季度日均新增合约数量超过 1000。

以太坊智能合约数量与日俱增，并受到越来越多的关注及运用，而其安全问题也随之暴露，攻击者利用安全漏洞对智能合约进行攻击，导致数字资产发生丢失或被盗取。故加强区块链智能合约的安全性随着以太坊合约的增加逐步进入大众视野，成为了区块链智能合约开发中工作中的一个难题。

区块链安全研究中心由此对区块链智能合约安全检测平台（www.sjtubsrc.net）内 31276 份智能合约进行检测，得出智能安全合约在九大安全漏洞的类型分布并对每种类型的漏洞严重等级进行评级。希望能加强区块链行业从事着对智能合约安全漏洞的了解，引起相关人群对智能合约安全的重视，促进区块链行业发展。

区块链安全研究中心 BSRC 由上海掌御信息科技有限公司与上海交通大学网络空间安全学院、中国信息通信研究院泰尔终端实验室共同组建。BSRC 致力于区块链安全领域的基础研究，开展包括但不限于区块链安全技术研发、安全行业标准制定、区块链应用场景安全研究、区块链金融应用合规性研究等工作。

上海交通大学网络空间安全学院（原信息安全工程学院），是由国家教育部、科技部、上海市政府和上海交通大学共同建设的国内首家学院建制的国家信息安全专业人才培养基地，拥有一支包括教授、副教授、兼职教授等 60 余名青年骨干教师和 200 多名博士、硕士研究生构成的高水平科研学术团队，先后承担了 300 余项国家重要

科研项目及横向项目，取得了包括国家科技进步二等奖等一系列重要科研成果，并参与国家一系列重要的信息安全标准制定和法规建设等。

中国泰尔实验室(CTTL)始建于 1981 年，行政隶属于工业和信息化部中国信息通信研究院(CAICT)，由工业和信息化部和国家质量监督检验检疫总局授权设立。实验室经历了不断的发展和融合，现在的实验室是由工业和信息化部中国信息通信研究院所属的电信传输研究所、通信计量中心、邮电工业标准化研究所和保定泰尔通信设备抗震研究所通过业务重组和资源整合而组成的。中国泰尔实验室是集信息通信技术发展研究，信息通信产品标准、测试方法、通信计量标准、计量方法研究，国内外产品的测试、验证、技术评估、测试仪表计量、通信软件的评估、验证为一体的高科技组织。

上海掌御信息科技有限公司是专业的区块链安全服务提供商，由国际顶尖的白帽子技术团队和密码学博士组成，曾是银联云闪付底层白盒密码引擎的技术供应商，也参与了多个国家标准和行业标准的制定，并与工信部中国泰尔终端实验室、上海交通大学网络空间安全学院共同成立了区块链安全研究中心，致力于区块链基础链、智能合约、应用客户端的黑盒安全测试和白盒代码审计，同时也提供加密数字热钱包和冷钱包的底层安全解决方案和身份验证方案。

中国区块链应用研究中心、杭州加密谷区块链科技有限公司和上海淳粹文化传媒有限公司是本白皮书联合发布方。

中国区块链应用研究中心是公益机构，由互联网金融博物馆联合部分区块链业界的领袖机构成立，其宗旨是与监管机构密切合作，共同推动区块链行业的培训认证和规范发展，鼓励区块链在实体经济中的场景应用，防范金融风险，促进中国区块链业界与全球同行的交流，

建立行业规则。

杭州加密谷区块链科技有限公司加密谷 Live（CryptoValley Live）是具有全球视野的区块链新媒体品牌，关注全球加密经济产业趋势，以图文、视频、直播、会议等形式报道全球区块链前沿资讯与深度思考。加密谷在瑞士楚格、旧金山、上海、香港、东京均设有记者站，运营中心设立在上海。

上海淳粹文化传媒有限公司（简称：淳粹传媒）以数据为基础，以知识为核心，以媒介为渠道，借助数知媒外部数据集成服务平台和资源管理系统，通过数据应用和资源整合为客户提供核心人物 IP 打造、风险与危机管理等整体解决方案。

目录

| 1 | 概述

|1.1| 以太坊介绍

以太坊（英文 Ethereum）是一个开源的有智能合约功能的公共区块链平台，目前已经发展成为仅次于比特币的全球第二大区块链，通过其专用加密货币以太币(Ether)提供去中心化的虚拟机 EVM(“以太虚拟机” Ethereum Virtual Machine）来处理点对点合约，基于以太坊的智能合约是全球最主流的智能合约。

以太坊虚拟机（EVM）是以太坊的核心，EVM 可以执行任意算法复杂度的代码。在计算机科学的术语中，以太坊是图灵完备的。开发者可以使用语法上类似 JavaScript 和 Python 的编程语言（Solidity）创建运行于 EVM 上的应用程序。

以太坊虚拟机（EVM）使用了 256 比特长度的机器码，是一种基于堆栈的虚拟机，用于执行以太坊智能合约。由于 EVM 是针对以太坊体系设计的，因此使用了以太坊账户模型（Account Model）进行价值传输。

EVM（以太坊虚拟机）的特性如下：

1、是一个完全基于栈结构的虚拟机

2、无寄存器的简单虚拟机

3、每个字节为 256 位长度

4、由以太坊黄皮书作为规范

5、每一步操作都需要花费一定的 gas

|1.2| 智能合约介绍

智能合约是一种旨在以信息化方式传播、验证或执行合同的计算机协议。智能合约允许在没有第三方的情况下进行可信交易，这些交易可追踪且不可逆转。

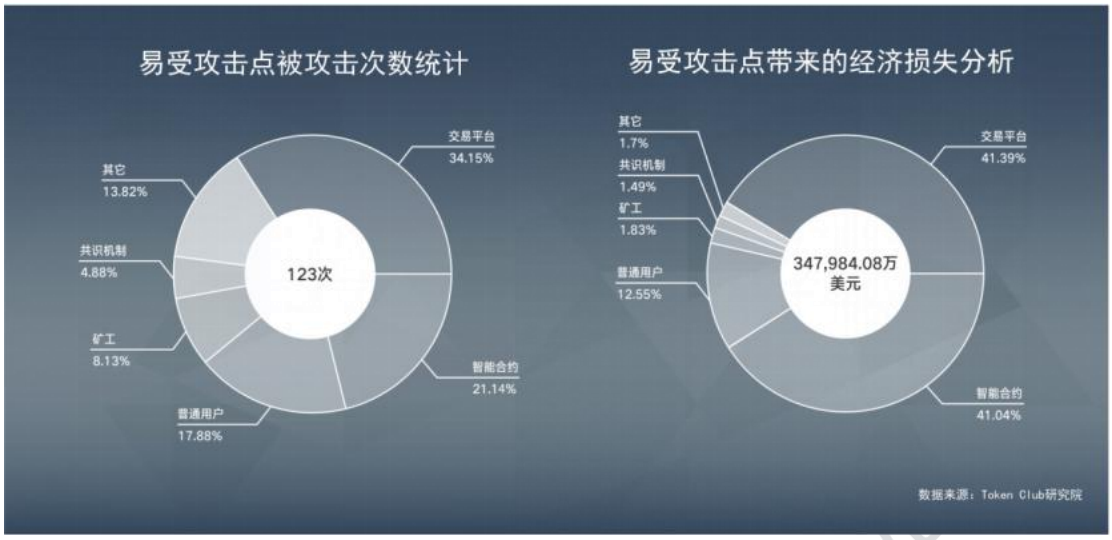
基于区块链技术的智能合约不仅可以发挥智能合约在成本效率方面的优势，而且可以避免恶意行为对合约正常执行的干扰。将智能合约以数字化的形式写入区块链中，由区块链技术的特性保障存储、读取、执行整个过程透明可跟踪、不可攥改。同时，由区块链自带的共识算法构建出一套状态机系统，使得智能合约能够高效地运行。但是，区块链上的所有用户都可以看到基于区块链的智能合约，这会导致包括安全漏洞在内的所有漏洞都可见，并且可能无法迅速修复。

|1.3| 以太坊智能合约安全风险

目前，以太坊智能合约的编程语言 solidity 和以太坊智能合约运行的虚拟环境 EVM 的设计还不完善，不排除出现安全漏洞的情况。

如果智能合约开发者稍有疏忽或者测试不充分，就有可能造成智能合约的代码存在漏洞。这对项目安全来说就像一颗隐藏的炸弹，一旦爆炸，后果将不堪设想。目前以太坊智能合约的安全漏洞容易导致用户资产贬值，被冻结，被非法转移等重大问题。

根据 Token Club 研究院对近年来区块链安全事件统计数据显示，易受攻击点被攻击次数共计 123 次，其中智能合约占比 21.14%，易受攻击点带来的经济损失共计 347984.08 万美元，其中智能合约占比 41.01%。重大安全事件数量与安全事件造成的经济损失也在 2018 年到达一个新值。



| 2 | 智能合约安全检测

|2.1| 智能合约安全检测内容说明

本白皮书是区块链安全研究中心以区块链智能合约安全检测平台（www.sjtubsrc.net）内 31276 份智能合约为数据样本以以太坊 ERC20 Token 标准进行检测，检测技术采用区块链智能合约安全检测平台的快速扫描引擎，检测时间为 2 周，最终得出智能安全合约在 call 安全漏洞、条件竞争漏洞、重入攻击、权限控制漏洞、数值溢出、事务顺序依赖、账户冻结及绕过、逻辑设计缺陷、错误使用随机

数等九大安全漏洞类型的分布，并对每种类型的漏洞严重等级进行了评级。

检测的智能合约相关数据如下：

- 检测智能合约数:31276
- 检测智能合约代码行数:9407593
- 检测智能合约函数数：371655
- 检测智能合约的交易笔数：87608190
- 检测智能合约价值:

|2.2| 智能合约安全漏洞分类与评级

智能合约安全漏洞及其严重等级定义如下：

类型	名称	严重等级
call 安全漏洞	DAOMethodCall	3
	DelegateCallWithUserInput	3
	UnsafeCallTarget	1
条件竞争漏洞	TODAmount	2
	TODReceiver	2
	TODTransfer	2
重入攻击	DAO	3
	DAOConstantGas	2
权限控制漏洞	UnprivilegedSuicide	3
	UnrestrictedEtherFlow	3

	UnrestrictedWrite	3
	UseOfOrigin	2
数值溢出	IntegerOverflow	3
事务顺序依赖	DivisionBeforeCallvalue	2
	DivisionBeforeMultiply	2
冻结账户绕过	LockedEther	3
逻辑设计缺陷	MissingInputValidation	2
	UnhandledException	2
	WriteOnly	2
错误使用随机数	UnsafeDependenceOnBlock	2
	UnsafeDependenceOnGas	2

|2.3| 快速扫描引擎介绍

快速扫描引擎应用在区块链安全研究中心推出的国内首个智能合约自动化检测系统 BSCSCS（在线平台 www.sjtubsrc.net）目前已经上线。利用快速检测引擎平台平均一分钟内可以完成一个智能合约的自动化检测，并公布所有的潜在风险点、漏洞细节以及相关的代码位置。平台现已面向所有用户开放注册和测试。

快速扫描引擎为基于事实推理的符号执行。本引擎根据 EVM 的特性创建了一套自定义的约束语句、根据当前已存在的常见合约漏洞创建了一套自定义的约束规则，通过对导入的智能合约代码进行约束转换、约束校验等，来判断其中的某些数据流是否能被证明安全或不安

全。

除快速检测引擎外，BSCSCS 平台同时还上线了基于符号执行的深度检测引擎。该引擎可以同时针对开源和未开源的智能合约代码进行安全审计，使得 BSCSCS 成为全球首个拥有双引擎的智能合约自动化审计系统。

BSCSCS 支持检测以太坊（Ethereum）上所有的智能合约，包括以太坊上超过半数未开源的 OPCODE 智能合约。如何针对这些合约进行审计并判断其是否安全可用一直是整个区块链行业的难点和痛点。BSCSCS 深度检测引擎平均半小时即可完成一个智能合约的深度安全审计。该项服务目前仅针对付费用户开放，测试完成后会自动发送审计报告到用户邮箱，报告中会体现所有的潜在漏洞以及位置。

此外，BSCSCS 还启用了智能合约自动化评分体系，通过多个数据维度对智能合约的安全性进行评分，目前，已完成了约 4 万个智能合约的自动化检测和评分。

|2.3.1| BSCSCS 快速扫描引擎的验证流程

1、通过智能合约反编译器将 EVM 虚拟机中的 opcode 反编译为 solidity 代码，并将每一句作为一个逻辑原语存储起来。

```
00: push 0x04
02: dataload
03: push 08
05: jump
06: jumpdest
07: stop
08: jumpdest
09: push 0x00
```

```
// entry
L1 a = 0x04
L2 b = dataload(a)
L3 ABI_9DA8(b)
L4 stop()
```

2、针对每一条具有不同属性、每一组具有不同顺序和语义关联的逻辑原语，使用转换引擎自动将其转换为一段由自定义约束语句组成的逻辑代码。

```
// entry
L1 a = 0x04
L2 b = dataload(a)
L3 ABI_9DA8(b)
L4 stop()
```

```
MustFollow(L2, L1)
VarTag(a, const)
MayDepOn(b, dataload)
Eq(c, 0x00)
```

3、调用自定义的逻辑约束集，对转换后的逻辑代码进行约束检查，如果有约束语句与逻辑约束集中的规则匹配，则该条约束语句被判定为存在该规则对应的漏洞。

```
MustFollow(L2, L1)
VarTag(a, const)
MayDepOn(b, dataload)
Eq(c, 0x00)
...
...
```

```
!MayDepOn(c, Caller) && !MayDepOn(b, Caller)
```

4、通过违反规则的语句找到相应的源代码中产生漏洞的语句。

```
1    L1 a = 0x04
2    L2 d = dataload(a)
3    ...
4    L3 stop()
5    ...
6    ...
7    L4  c = 0x00
8    L5  sstore(c, b);
9    ...
```

|2.3.2| BSCSCS 快速扫描引擎支持扫描漏洞种类

快速扫描引擎当前可支持以太坊合约大部分漏洞种类的智能扫描，包括：条件竞争漏洞、重入攻击、权限控制漏洞、数值溢出、顺序依赖、账户冻结、逻辑缺陷、随机数问题、Call 函数问题等。

| 3 | 安全检测结果及分析

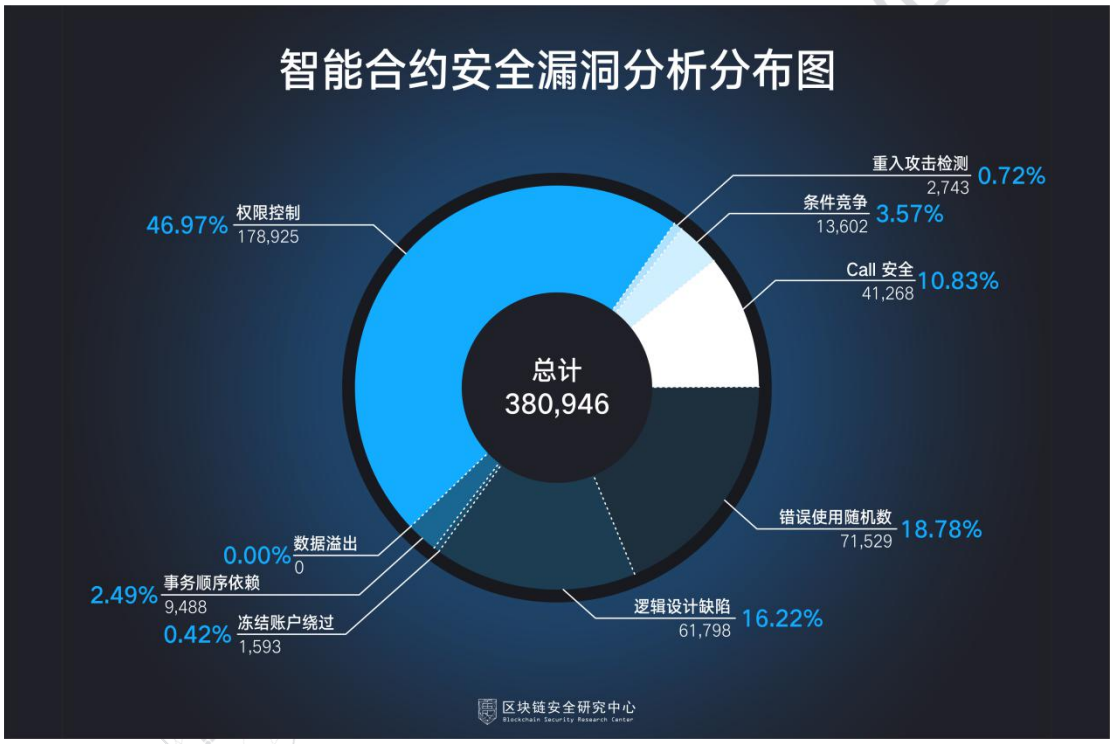
|3.1| 安全漏洞分布

|3.1.1| 智能合约安全漏洞类型分布图

根据检测结果，按照安全漏洞类型智能合约安全漏洞的分布如下：

类型	数量	占比
call 安全	41268	10.83%
条件竞争	13602	3.57%
重入攻击检测	2743	0.72%

权限控制	178925	46.97%
数值溢出	0	0.00%
事务顺序依赖	9488	2.49%
冻结账户绕过	1593	0.42%
逻辑设计缺陷	61798	16.22%
错误使用随机数	33809	10.38%

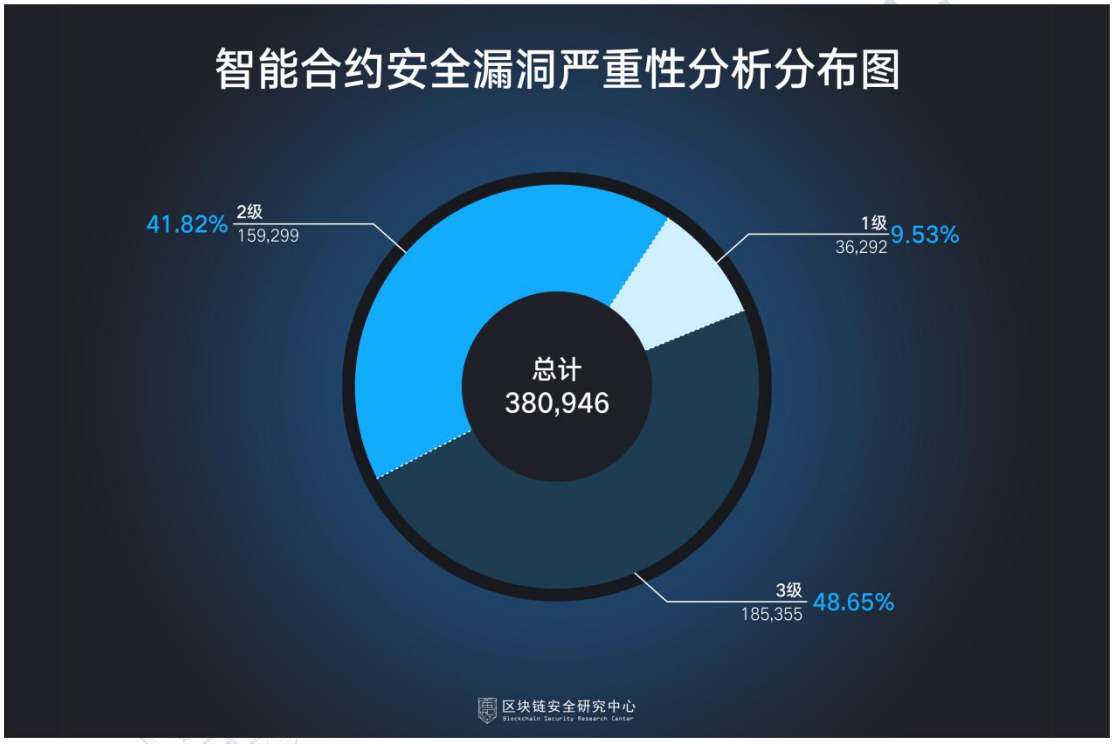


智能合约安全漏洞目前有 9 类，从安全漏洞的数量来看，权限控制占比最重，达到了 46.97%，远高于其他类型。而数据溢出未在本次数据样本中被监测出。

|3.1.2| 智能合约安全漏洞严重等级分布图

根据检测结果，按照安全漏洞严重等级智能合约安全漏洞的分布如下：

漏洞严重等级	数量	占比
1 级	36292	9.53%
2 级	159299	41.82%
3 级	185355	48.65%



从安全漏洞的严重性看，3 级漏洞占比最重，高达 48.65%，二级漏洞也有 41.82%。

|3.2| call 安全漏洞

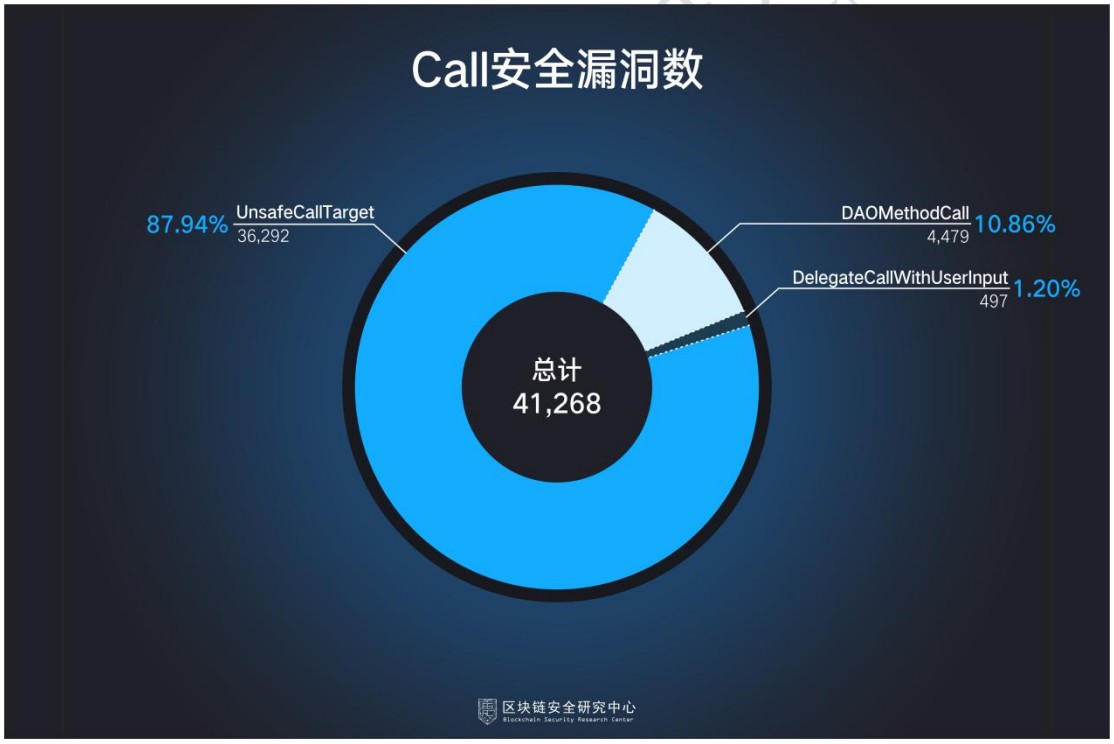
call 安全漏洞：

1. DAOMethodCall：描述了一种更抽象的重入漏洞，指目标合约

在进行外部合约调用后，发生相应状态的改变，因此存在重入漏洞的可能。建议：1）对调用请求的参数内容进行检测；2）确保调用前改变合约状态；3）引入互斥锁。

2. DelegateCallWithUserInput：DelegateCall 属于一种标准信息调用，但相关代码会在调用合约的上下文中运行，当 Call 的内容由用户自定义时可能会让攻击者可以借助部分已有函数对当前合约存储位置进行任意修改。建议构建库合约时使用 library 关键字。

3. UnsafeCallTarget：当合约调用不受信任的外部地址时，可能会被攻击者利用以执行恶意内容。因此需要提前对调用地址和传输的数据内容做相应检查。



典型案例：

DelegateCall漏洞导致的安全事件：2017年11月Parity钱包遭到攻击，导致2.85亿美元的以太币被冻结。由于自2017年7月20日开始部署的多重签名钱包技术出现安全问题，共用的library合约被抹除，

导致以太坊钱包Parity的多重签名钱包的余额都被冻结，无法移动。最终导致有930000个以太坊（价值2.8亿美元）不能移动了。

UnsafeCallTarget漏洞导致的安全事件：2018年5月30日，Adrian Manning博士在Solidity Security: Comprehensive list of known attack vectors and common anti-patterns一文[1]中公布了他发现的一批可重入钓鱼合约，这些合约通过在构造函数中使用而已合约代替期望的合约，可以让试图使用重入漏洞的攻击者损失Ether。

|3.3| 条件竞争漏洞

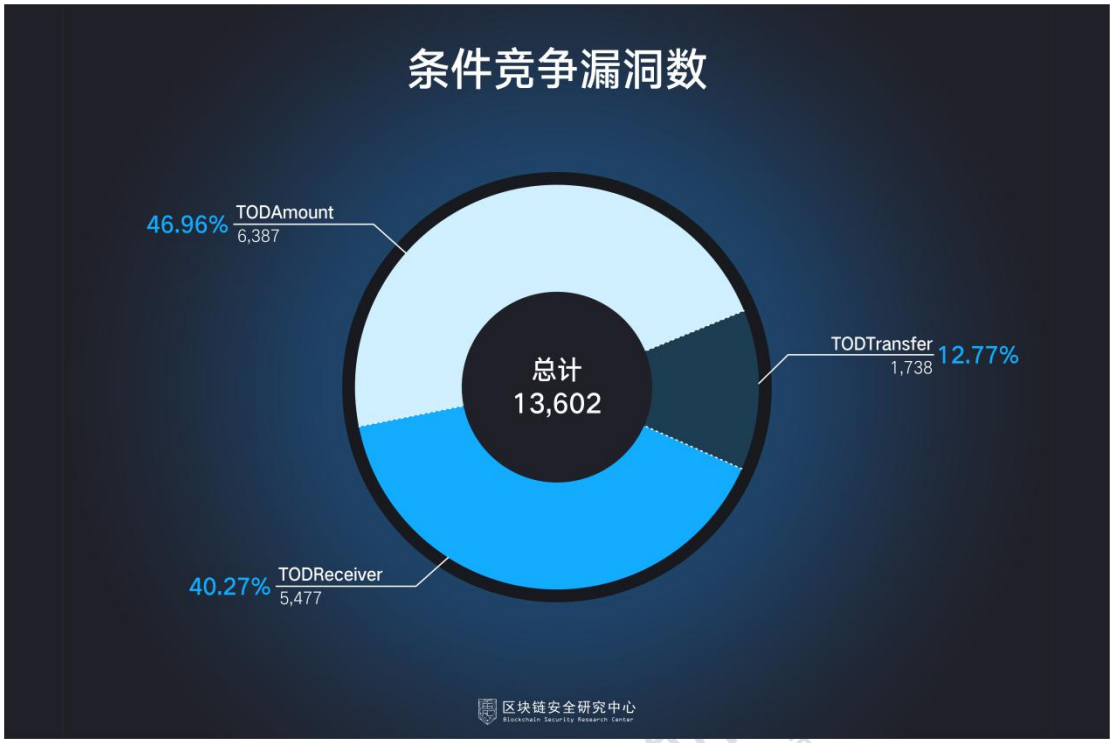
条件竞争漏洞：

1. TODReceiver：合约中的条件竞争漏洞可能对代币接收者造成影响。每次普通用户尝试申请调用第三方合约的 transfer 方法进行转账时，都有可能由于条件竞争漏洞的存在将代币转到攻击者账户上。建议针对接收者账户进行校验，必须与调用者地址相符才能完成转账。

2. TODAmount：合约中的条件竞争漏洞可能对传输的代币数量造成影响。当用户请求转移不定量代币时，有可能由于条件竞争漏洞问题将更多的代币转移到攻击者账户上。建议针对传输的代币数量进行检查，防止其在传输前被他人篡改。

3. TODTransfer：合约中的条件竞争漏洞可能对传输顺序造成影响。建议在逻辑中设置 gas 上限，防止恶意用户由于 gas 太高获得优先。

[1] 来源：https://blog.sigmaprime.io/solidity-security.html



典型案例：

TODTransfer 漏洞导致的安全事件：2018 年 6 月 15 日 Armors 团队发现 ERC20 标准的 approve 方法存在一个潜在的条件竞争漏洞，允许攻击者在合约使用 approve()时，根据 gasPrice 的高低改变交易达成的顺序，从而可能获得更多的 Ether[2]。

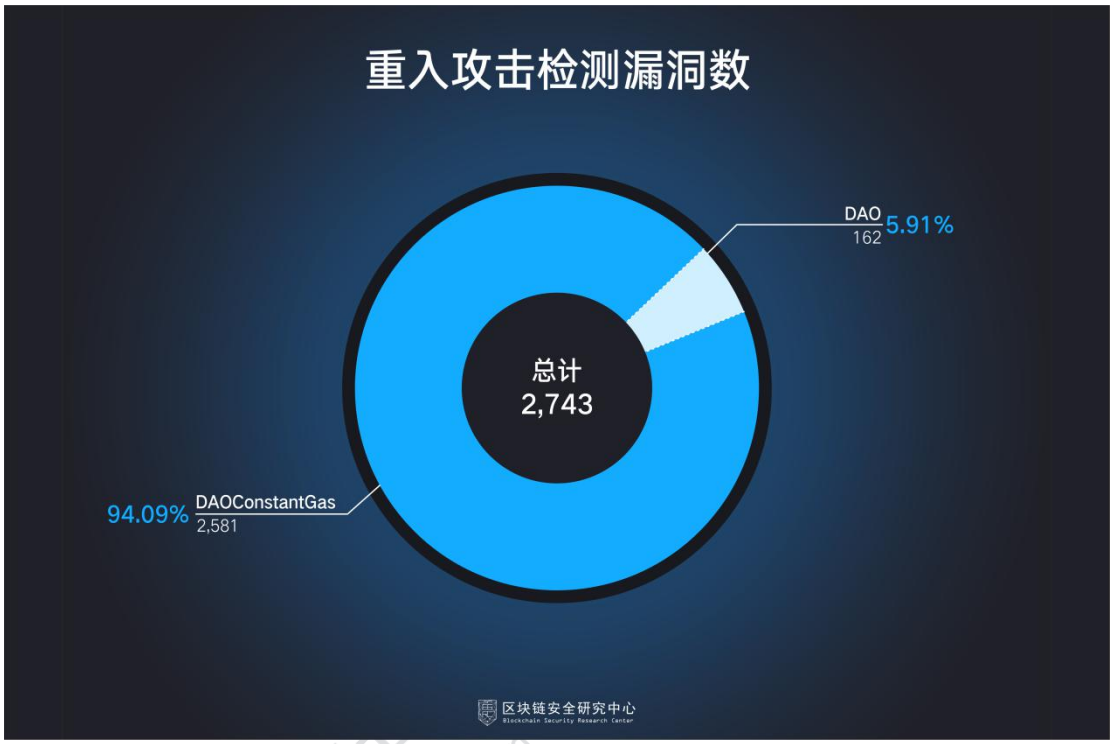
|3.4| 重入攻击漏洞

重入攻击漏洞：

1. DAO：漏洞发生至少有如下条件：1）转账操作发生在扣款之前；2）转账操作所拥有的 gas 量能发起多次转账请求。建议：1）使用 transfer 函数；2）确保 transfer 之前改变合约状态；3）引入互斥锁，将一次转账作为一个原子操作进行。

[2] 来源：https://mp.weixin.qq.com/s/bVf4MZLpmmTZ4xwNDq97SQ

2. DAOConstantGas：合约所发送的 gas 量固定(如使用了 send 或 transfer)，但转账发生在扣款前。建议：1）使用 transfer 函数；2）确保 transfer 之前改变合约状态；3）引入互斥锁，将一次转账作为一个原子操作进行。



典型案例：

DAO漏洞导致的安全事件： 2016年6月17日The DAO组织编写的智能合约中漏洞被黑客利用，区块链业界最大的众筹项目TheDAO（被攻击前拥有1亿美元左右资产）遭到攻击，导致300多万以太币资产被分离出TheDAO资产池。TheDAO编写的智能合约中有一个splitDAO函数，攻击者通过此函数中的漏洞重复利用自己的DAO资产来不断从TheDAO项目的资产池中分离DAO资产给自己。

|3.5| 权限控制漏洞

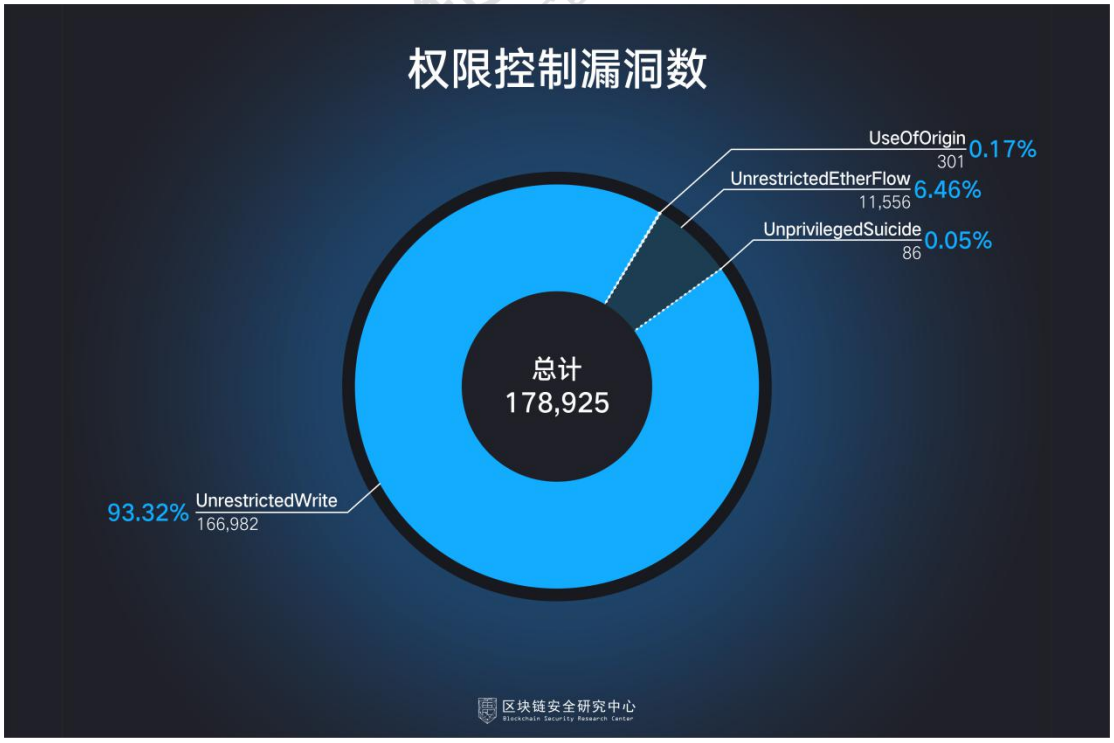
权限控制漏洞：

1. UnprivilegedSuicide：合约自毁时会将合约自身删除，并返还其所保存的所有账户余额，但如果没有做好权限控制漏洞，可能导致任意用户都能进行自毁操作。建议设置为仅 owner 用户有自毁权限。

2. UnrestrictedEtherFlow：转账行为往往应该由一组具有特定权限的用户来操作，如果转账操作不受任何权限限制，则可能带来潜在的问题。建议对转账操作添加权限修饰。

3. UnrestrictedWrite：指一个合约中存储的参数值(如 owner)能不受权限限制地被任意改写。建议对于具有修改功能的函数添加相应的权限修饰符。

4. UseOfOrigin：tx.origin 变量会遍历调用栈并返回初始调用者的地址，在使用 tx.origin 的场景中，合约比较容易受到类似钓鱼的攻击。建议在授权时不使用 tx.origin。



典型案例：

UnrestrictedEtherFlow 漏洞导致的安全事件:2017 年 7 月 19 日，在 Parity 钱包第一次被黑事件中，多签名钱包调用的库合约中存在权限控制漏洞不当的初始化函数，最终导致合约所有权被重置为攻击者地址，造成 15 万以太币约 3000 万美元被盗。

|3.6| 数值溢出漏洞

数值溢出漏洞：

1. IntegerOverflow：当合约在进行算术运算时，若没有对上下限进行约束，可能会存在溢出，导致某些账户中出现大量代币，也可能导致代币价值归零。建议使用 SafeMath 库进行安全的算术操作。

典型案例：

IntegerOverflow 漏洞导致的安全事件：

1. 2018 年 4 月 22 日中午，黑客利用以太坊 ERC-20 智能合约中 BatchOverFlow 漏洞中整数溢出的漏洞攻击美链 BEC 的智能合约，成功地向两个地址转出了天量级别的 BEC 代币，导致市场上海量 BEC 被抛售，该数字货币价值几近归零，给 BEC 市场交易带来了毁灭性打击。

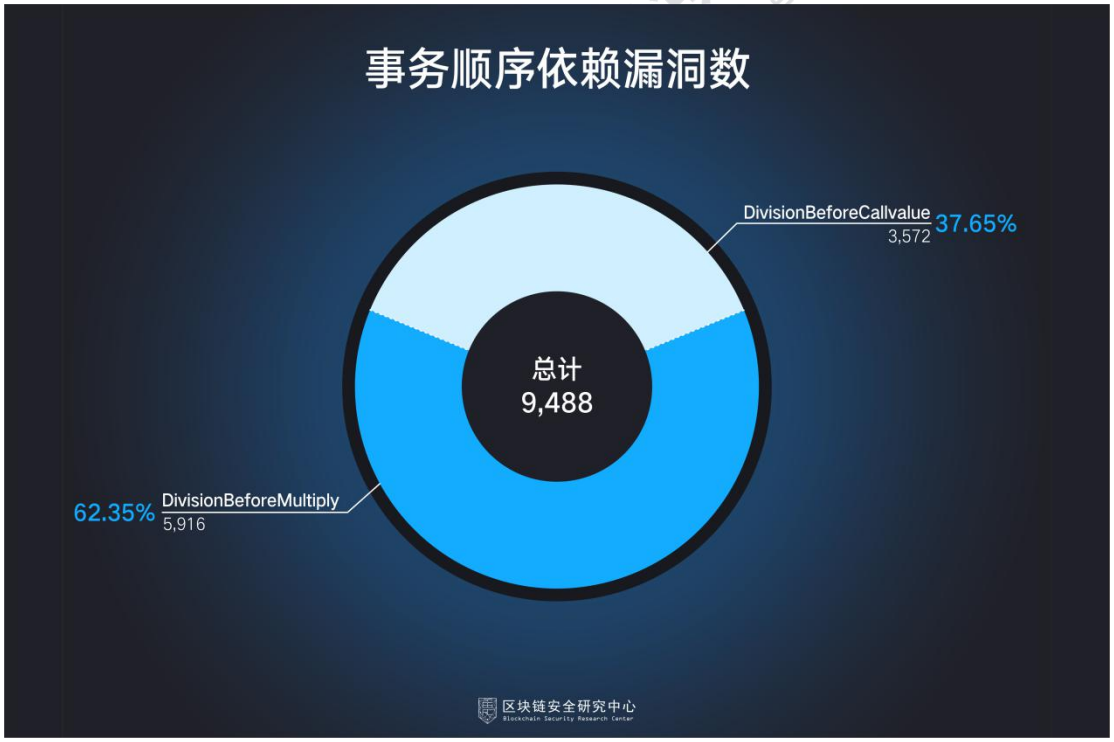
2. 2018 年 7 月 25 日，EOS Fomo 3D 游戏合约遭到溢出攻击，致使奖励池内累积的 94000 个 EOS 变成负数。次日，该游戏开发团队声称，6 万个 EOS 被攻击者套现。这个 EOS 版本的 Fomo3D 游戏在上线 4 天后宣告终结。

|3.7| 事务顺序依赖漏洞

事务顺序依赖漏洞：

1. DivisionBeforeMultiply：由于 solidity 中数字的唯一类型只有整形，整数除法的结果会被四舍五入到最近的整数，因此如果先做除法后做乘法可能会带来不可预知的后果。建议使用除法前仔细检查算术逻辑，确认不会受到除法四舍五入的影响。

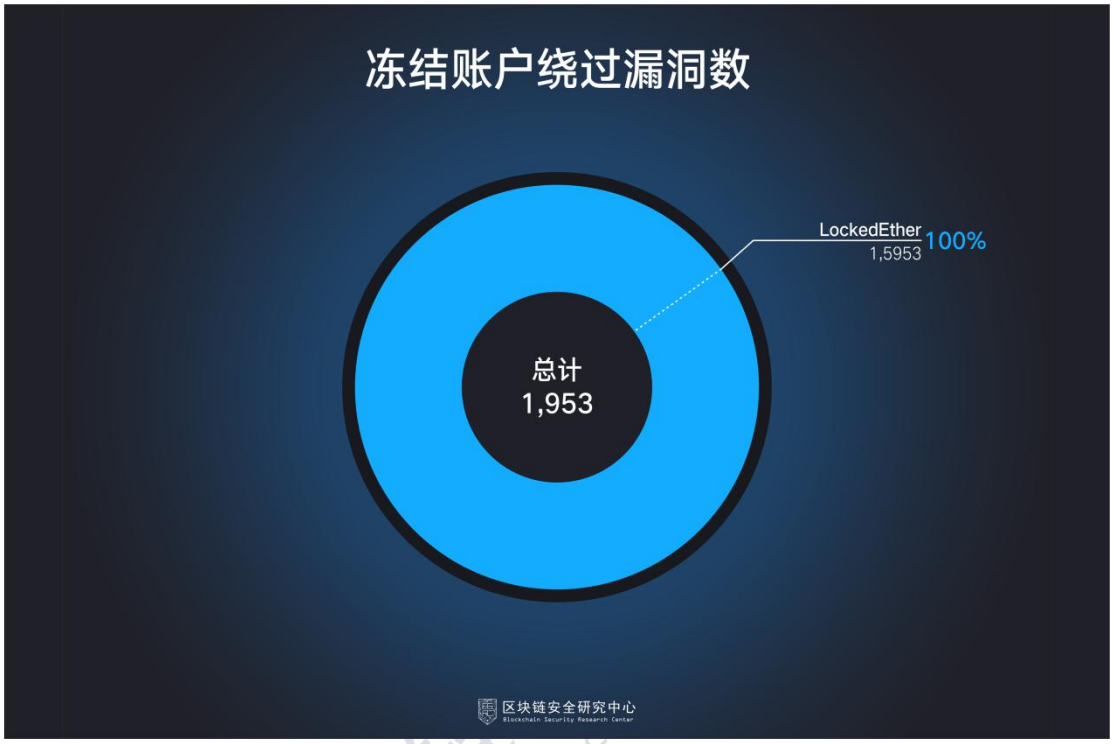
2. DivisionBeforeCallvalue：由于 solidity 中数字的唯一类型只有整形，整数除法的结果会被四舍五入到最近的整数，因此如果发送的代币数量是经过除法得出的结果，就需要格外注意。建议使用除法前仔细检查算术逻辑，确认不会受到除法四舍五入的影响。



|3.8| 账户冻结及绕过漏洞

账户冻结及绕过漏洞：

1. LockedEther：以太冻结问题指合约能正常接收用户存入的以太，但无法正常取出。建议做好函数存取款逻辑的检查。



|3.9| 逻辑设计缺陷漏洞

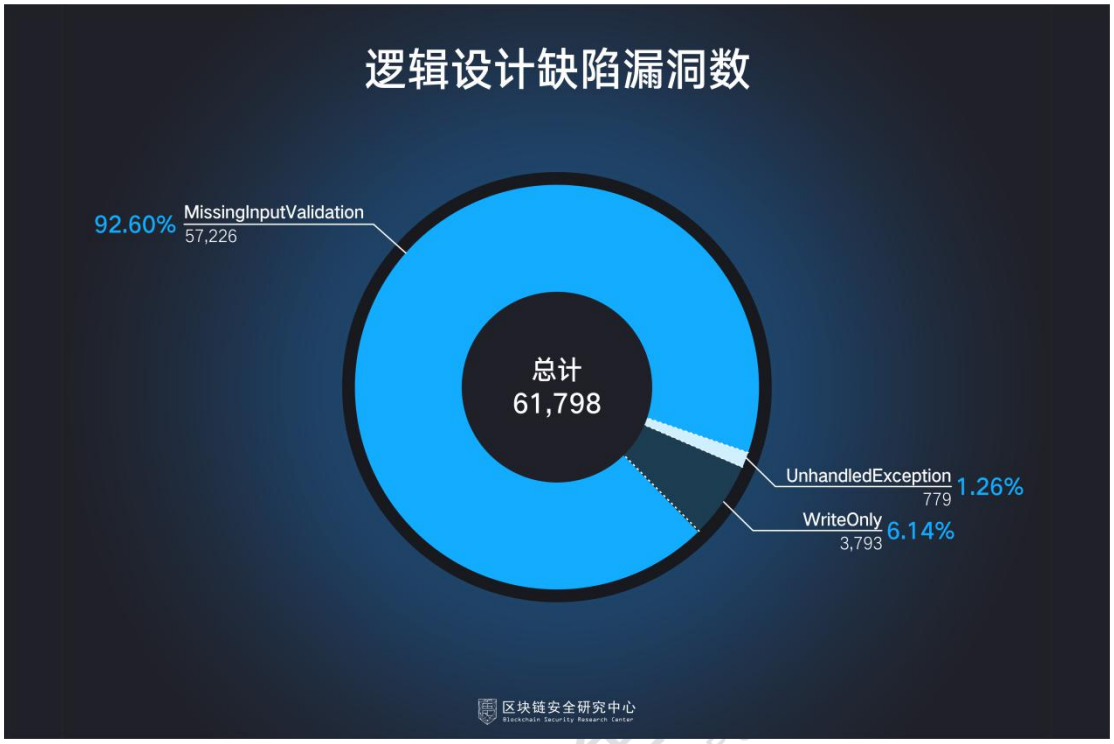
逻辑设计缺陷漏洞：

1. MissingInputValidation：不受控制的函数传入值可能会带来意料之外的操作结果。建议在使用用户传入的参数之前首先进行相应的检测。

2. UnhandledException：当 call 和 send 方法在执行过程中碰到 gas 不足或溢出等情况时会返回一个非零的值。建议使用 require 检测异常。

3. WriteOnly：构建合约中开发者经常会使用到一些与合约逻辑

无关的参数，若需要对这些参数内容做改动，建议配合 events 进行。



典型案例：

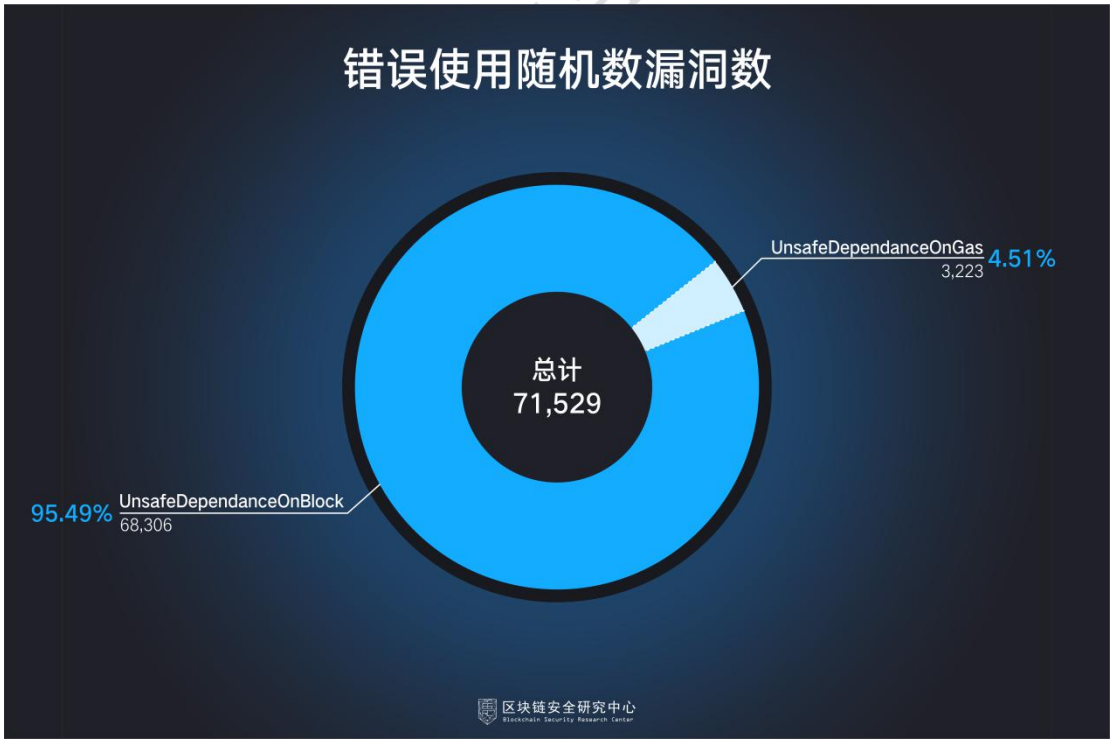
UnhandledException漏洞导致的安全事件：2016年2月6日至8日，The King of the Ether Throne “纷争时代”游戏中，玩家需要发送给合约一些以太币，从而获得“王位”。只要拿到了王位，玩家就会被加到皇庭，并且永远地被记录在区块链上，支付的数额由现任国王决定。很显然，当前国王可以通过买卖国王获得利润。当一个用户声称为国王后，合约就发送赔偿金给现任国王，并指定这个用户为新的国王。然而，这个合约并没有检查send函数的不成功调用，这样一旦合约在执行过程中产生了异常，现任国王就有可能同时失去王座和赔偿金，最终许多游戏中的退位君王的补偿和未接受款项无法退回用户玩家的钱包，从而导致了“庞氏陷阱”。

|3.10| 错误使用随机数漏洞

错误使用随机数漏洞：

1. UnsafeDependenceOnBlock：由于 solidity 很难实现随机数的生成，且以太坊中很多合约都开源，因此使用自创的随机数算法和自定义 seed（如使用区块的相应信息）会很容易被预测到随机数的内容。建议使用较为安全的伪随机数产生方法，如 Oraclize、BTCRelay、Signidice、Commit－reveal approach 等。

2. UnsafeDependenceOnGas：由于 solidity 很难实现随机数的生成，且以太坊中很多合约都开源，因此使用自创的随机数算法和自定义 seed（如使用 gas 的数量）会很容易被预测到随机数的内容。建议使用较为安全的伪随机数产生方法，如 Oraclize、BTCRelay、Signidice、Commit－reveal approach 等。



典型案例：

UnsafeDependenceOnBlock 漏洞导致的安全事件：2018 年 1 月 31

日,Arseny Reutov 在分析了 3649 份使用某种伪随机数发生器(PRNG)的实时智能合约后，发现了 43 份可被利用的合约漏洞[3]。

| 4 | 典型分析案例

我们从检测样本合约中选取了 TopToken 智能合约进行分析，TopToken 智能合约以太坊地址为：0x0E6BB94B7f25B96f13E0baf5bC04b8Ba39b897A8。

此智能合约的源代码可在 https://etherscan.io/address/0x0e6bb94b7f25b96f13e0baf5bc04b8ba39b897a8#code 上查看。

通过 BSCSCS 平台的快速扫描引擎，TopToken 的安全漏洞扫描结果如下：

类型	名称	严重等级	安全漏洞数
call 安全	DAOMethodCall	3	0
	DelegateCallWithUserInput	3	0
	UnsafeCallTarget	1	0
条件竞争	TODAmount	2	0
	TODReceiver	2	0
	TODTransfer	2	0
重入攻击检测	DAO	3	0
	DAOConstantGas	2	0

[3] 来源：https://blog.positive.com/predicting-random-numbers-in-ethereum-smart-contracts-e5358c6b8620

权限控制	UnprivilegedSuicide	3	0
	UnrestrictedEtherFlow	3	0
	UnrestrictedWrite	3	7
	UseOfOrigin	2	0
数值溢出	IntegerOverflow	3	0
事务顺序依赖	DivisionBeforeCallvalue	2	0
	DivisionBeforeMultiply	2	0
冻结账户绕过	LockedEther	3	0
逻辑设计缺陷	MissingInputValidation	2	9
	UnhandledException	2	0
	WriteOnly	2	0
错误使用随机数	UnsafeDependenceOnBlock	2	14
	UnsafeDependenceOnGas	2	0

根据漏洞分布，我们可以看出 TopToken 安全漏洞主要体现在权限控制、逻辑设计缺陷和错误使用随机数三个方面。

按照漏洞数量排序：错误使用随机数、逻辑设计缺陷和权限控制。

按照漏洞严重等级：L2、L3，其中 L2 安全漏洞数量为 23 个，L3 安全漏洞数量为 7 个。

此合约的检测结果可通过 BSCSCS 平台进行查看，地址为：http://www.sjtubsrc.net/detail.html?r=0x0E6BB94B7f25B96f13E0baf5bC04b8Ba39b897A8 。

| 5 | 结束语

基于区块链技术的智能合约不仅可以发挥智能合约在成本效率方面的优势，而且可以避免恶意行为对合约正常执行的干扰。将智能合约以数字化的形式写入区块链中，由区块链技术的特性保障存储、读取、执行整个过程透明可跟踪、不可攥改。同时，由区块链自带的共识算法构建出一套状态机系统，使得智能合约能够高效地运行。但是，区块链上的所有用户都可以看到基于区块链的智能合约，这会导致包括安全漏洞在内的所有漏洞都可见，并且可能无法迅速修复。

但由于区块链是新兴技术,开发人员和安全人员都很缺乏,尤其在传统安全领域人才稀缺的情况下,区块链安全人才更是一人难求。这一现状导致的直接结果是,所谓“开发人员”都直接从网上下载大量智能合约代码,修改后便直接使用。但这些源自网络的代码本身可能存在严重漏洞,经过不断复制后,也把安全漏洞扩散了。一旦遭人利用,个体问题便迅速扩散为群体性灾难事件。

那么这类安全事件该如何防范呢？作为智能合约的相关方又该如何避免漏洞的发生呢？

1.开发者应该提高自己的安全意识。现在发现的漏洞中，大多是因为直接使用普通的加减乘除符号,但却没有对可能溢出的情况作判断，这就造成了数据溢出的隐患，而解决方法也很简单，使用安全的运算库 library SafeMath 就可以彻底避免数据溢出的问题。

2.项目方也应建立自己统一的合约编写安全标准,并对照安全标准严格执行，进行逐一检查。在完成智能合约编写后，请专业的智能合约审计公司，对合约代码用形式化验证的方法进行审计，并由审计公司给出审计报告和潜在漏洞的修复建议。

3.数字资产交易平台也应该做好对项目方的审核工作和自身安全防护。对交易平台中的项目，应要求其能提供智能合约安全凭证，

避免有漏洞的代币对交易平台的信誉产生不良影响。

2018 年是区块链行业发展的转折年，随着数字资产市场逐渐转冷，区块链项目开始出现明显分化，这些都是行业泡沫逐渐稀释的迹象。人们越来越意识到，除了发币圈钱之外，区块链技术需要更多的应用场景来证明自己的价值。

随着区块链行业日趋发展，应用场景逐渐增多，区块链智能合约的安全问题也成为了区块链产业的一大重点。未来，人们在不断提高对区块链的理解和认知的同时，也要对智能合约的安全加以重视，对安全漏洞加以防范。