Dice2win 区块链博彩游戏庄家选择性中止漏洞

Zhiniang Peng from qihoo 360 core security

Dice2win 目前是以太坊上一款异常火爆的区块链博彩游戏。号称“可证明公平的”Dice2win 目前每日有近百万人名币的下注额，是总交易量仅次于 etheroll 的第二大以太坊博彩游戏。然而我们分析发现，dice2win 中的所有游戏都存在庄家选择性中止漏洞，庄家可以选择性公布中将结果从而导致用户无法获胜。

Dice2win 游戏介绍：

Dice2win 目前有包括“抛硬币”、“掷骰子”、“两个骰子”、“过山车”几款游戏。其介绍如图：

在这些游戏中，每个用户单独下注与庄家进行一对一对赌。游戏的本质，是用户和庄家在去中心化的以太坊智能合约平台上通过一系列协议来生成随机数。如果用户猜中随机数，则用户胜利，否则庄家胜利。

在进一步介绍 Dice2win 工作流程和公平性分析之前，我们先讨论一个历史悠久的密码学问题：Mental poker。 Mental poker 是由 Shamir, Rivest 和 Adleman 1978 年在文章“Is it possible to play a fair game of ‘Mental Poker”中首次提出的概念。(Shamir, Rivest 和 Adleman 有没有很眼熟？没错，就是你知道的 RSA)其本质是想解决：在没有可信第三方的参与的情况下(可信平台或软件)，两个不诚实的参与方如何在网络上进行一场公平的棋牌游戏。在公平性的定义中，有非常重要的一点：如果任何一方收到了游戏结果，那么所有的诚实方都应该收到结果。

Dice2win 实际上是利用区块链实现 mental poker 的典型案例。但我们发现，Dice2win 并不满足 mental poker 的公平性安全。本文的选择性中止攻击(selective abort attack)，实际上就是对公平性的一种攻击。

Dice2win 区块链博彩游戏选择性中止攻击

这里我们来看看 Dice2win 的工作原理。Dice2win 游戏的本质，是用户和庄家在去中心化的以太坊智能合约平台上通过一系列协议来生成随机数。如果用户猜中随机数，则用户胜利，否则庄家胜利。

1. 【庄家承诺】庄家(secretSigner)随机生成某随机数 reveal，同时计算 commit = keccak256 (reveal)对该 reveal 进行承诺。然后根据目前区块高度，设置一个该承

诺使用的最后区块高度 commitLastBlock。 对 commitLastBlock 和 commit 的组合体进行签名得到 sig，同时把(commit, commitLastBlock,sig)发送给玩家。

2. 【玩家下注】玩家获得(commit, commitLastBlock,sig)后选择具体要玩的游戏，猜测一个随机数 r，发送下注交易 placeBet 到智能合约上进行下注。
3. 【矿工打包】下注交易被以太坊矿工打包到区块 block1 中，并将玩家下注内容存储到合约存储空间中。
4. 【庄家开奖】当庄家在区块 block1 中看到玩家的下注信息后。则发送 settleBet 交易公开承诺值 reveal 到区块链上。合约计算随机数 random_number=keccak256(reveal,block1.hash)。如果 random_number 满足用户下注条件，则用户胜，否则庄家胜。此外游戏还设有大奖机制，即如果某次 random_number 满足某个特殊值(如 88888)，则用户可赢得奖金池中的大奖。

Dice2win 在官网和白皮书宣称自己的游戏具有数学上可证明的公平性，其随机数是随机数生成过程由矿工和庄家共同决定，矿工或者庄家无法左右游戏结果，所以玩家可以放心下注。此外，在一些介绍以太坊智能合约安全的文章中，我们也看到一些作者将 Dice2win 的随机数生成过程称为极佳实践。

然而我们分析发现，dice2win 中的所有游戏都会受到庄家选择性中止攻击，庄家可以选择性公布中将结果从而导致用户无法获胜或赢得彩票。我们考虑如下两个攻击场景：
场景 1：
用户下注额大，且赔率高得情况下。用户下下注产生 block1 后，block1.hash 实际上就已经固定了。此时庄家已经可以计算出 random_number，从而计算出用户的投注结果和盈亏。则庄家可以选择性中止交易。如果用户不中奖，则庄家公布正常开奖结果。如果用户中奖，则庄家可因为“网络用户和技术原因”从而导致用户该笔下注失效。

场景 2：
用户下注额不大，但是 block1 产生后庄家发现 random_number 导致用户中彩票。则庄家可以选择性中止交易，导致用户该笔下注失效。

在这两种攻击场景下，庄家都能够轻松控制交易结果。当然庄家并不会对每笔交易都发起这种攻击，而是可以选择用户获奖特别大的交易进行操控。Dice2win 官方实际上已经在智能合约代码得注释中声明了可能会发生“技术问题和以太坊拥堵”原因造成荷官无法开奖(大约 1 个小时内)，则用户可以提回下注款。

```
    // This is a check on bet mask overflow.
    uint constant MAX_BET_MASK = 2 ** MAX_MASK_MODULO;

    // EVM BLOCKHASH opcode can query no further than 256 blocks into the
    // past. Given that settleBet uses block hash of placeBet as one of
    // complementary entropy sources, we cannot process bets older than this
    // threshold. On rare occasions dice2.win croupier may fail to invoke
    // settleBet in this timespan due to technical issues or extreme Ethereum
    // congestion; such bets can be refunded via invoking refundBet.
    uint constant BET_EXPIRATION_BLOCKS = 250;
```

漏洞修复方案：

造成该漏洞的本质原因在于，该方案的随机数对于庄家而言并不是真正的随机。庄家可以提前知道下注的结果。想一想如果你去一个声称“绝对公平”的赌场赌骰子。在完成下注后，庄家是有一种方法先偷看一眼骰子结果，算一算盈亏之后再决定是否开奖(重摇)，这样的骰子，是真的随机的吗？

要修复该问题实际也很简单，只要让庄家在公布承诺之后才能看到随机数的值。以下为我们修正后的方案：

1. 【庄家承诺】庄家(secretSigner)随机生成某随机数 reveal1，同时计算 commit1 = keccak256 (reveal1)对该 reveal1 行承诺。然后根据目前区块高度，设置一个该承诺使用的最后区块高度 commitLastBlock。 对 commitLastBlock 和 commit1 的组合体进行签名得到 sig，同时把(commit1, commitLastBlock,sig)发送给玩家。
2. 【玩家下注，并承诺】玩家获得(commit1, commitLastBlock,sig)后选择具体要玩的游戏，猜测一个随机数 r。并生成 commit2 = keccak256 (reveal2)发送下注交易 placeBet((commit1, commitLastBlock,sig, commit2)到智能合约上进行下注。
3. 【矿工打包】下注交易被以太坊矿工打包到区块 block1 中，并将玩家下注内容存储到合约存储空间中。
4. 【庄家打开承诺】当庄家在区块 block1 中看到玩家的下注信息后。在限定时间内公开承诺值 reveal1 到区块链上，同时将 block1.hash 存储到合约存储中。
5. 【玩家打开承诺开奖】当玩家看到庄家 reveal1 消息后，在限定回合内公开 reveal2 到智能合约上调用 settleBet 开奖。合约计算 random_number=keccak256(reveal1，reveal2,block1.hash)计算得到真随机数。

注意在修正后的方案中，block1.hash 在此处是否可以略去，取决于具体的游戏场景。

后记：

本文中的 Dice2win 区块链博彩游戏选择性中止攻击实际上是一个非常明显的针对 mental poker 的攻击手段，要解决这个问题在密码学上也并非什么难事。我们无法断定该漏洞是否为官方有意留下。但这样明显的漏洞存在于目前最热门的区块链博彩游戏中，我们不禁要问一句：目前去中心化的区块链博彩游戏的真的公平吗？