EOS hard fail 状态攻击细节披露及修复方案

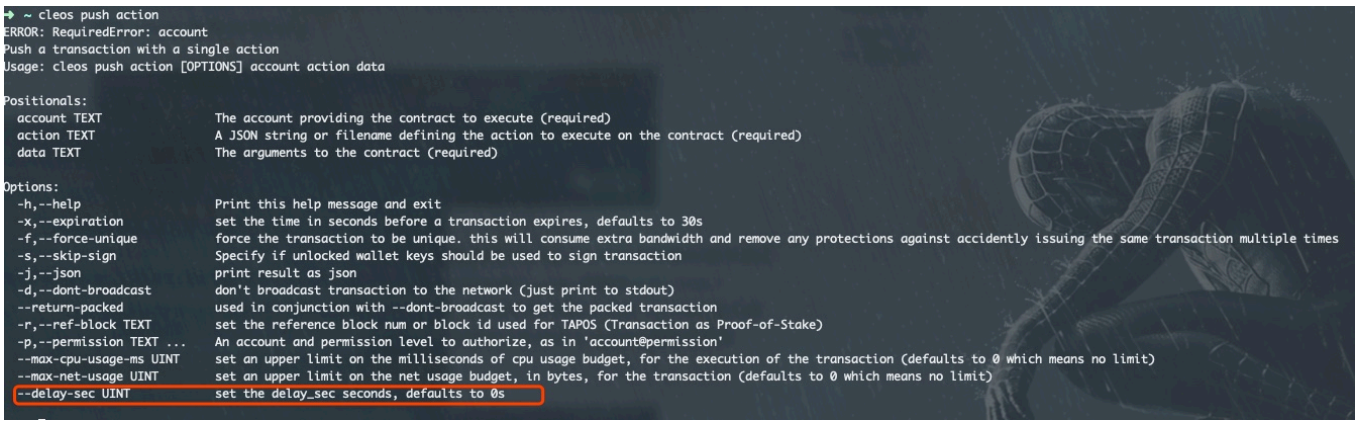
披露时间线

2019年3月10日，我们捕获了 EOS DApp 上的一种新型攻击手法，一个帐号名为 fortherest12 的攻击者通过 hard fail 状态攻击手法攻击了 EOS 游戏 Vegas town ，并造成了一定数量的损失。
2019年3月10日，我们注意到出现了数量更多的 hard fail 类型攻击。
2019年3月11日，我们披露了相关的攻击手法，但是没有披露具体的攻击细节，并及时联系了相关的交易所以及项目方
2019年3月12日，我们发布红色预警，提醒交易所和钱包需要对 EOS 交易执行状态进行校验，必要时可暂停充提系统。
2019年3月13日，细节正式公开。

漏洞细节

参考官方文档（https://developers.eos.io/eosio-nodeos/docs/how-to-monitor-state-with-state-history-plugin），可以得知 EOS 交易的执行状态存在多种,相应的类别及相应的描述分别为：
1、executed：交易正确执行，没有触发 error handler
2、soft_fail：交易客观失败（没有执行），但正确触发 error handler
3、hard_fail：交易客观失败，但没有触发error handler
4、delayed：交易延迟/deferred/在队列中待执行
5、expired：交易过期

此次的攻击手法就是利用了上述状态中的 hard_fail 状态进行攻击，在以前的开发过程中，许多开发者都不曾碰到过这种交易执行状态，常规的区块浏览器上也无法查询到相关的交易，造成了开发者缺少对这种交易状态的认知。开发中的惯常思维也是只有合约才能发起延时交易。但是，通过cleos中的具体参数配置delay-sec 参数：



即使使用非合约帐号也能正常发起 delay 交易，针对 使用中心化开奖的 DApp 项目方或使用中心化管理的交易所

及钱包，如果没有对 EOS 交易的执行状态进行校验，就有可能出现 “假充值” 攻击，攻击者不需要付出任何成本，却可以获得大量的 EOS。这是一种全新的攻击手法，而且也是大家容易十分忽略的点，但是导致的危害确是巨大的。对于这种手法，我们已经捕获到真实攻击，并成功阻止了一起损失金额以人民币亿为单位的攻击。

基于上述情况，我们于3月12日发布了我们的红色预警，但由于我们的翻译不够严谨，在 EOS 社区引起了恐慌，让 EOS 社区认为这是 EOS 的漏洞，我们对此致以歉意。EOS 社区的一些成员认为，只要交易所及钱包做好相关的检查，并不会出现此类问题。但是我们很难要求所有程序员都能写出最佳安全实践的代码，当出现不严谨的校验方式便会导致攻击发生。

经过我们的分析，我们更倾向于 transaction status 属于 EOS 的一种特性（features），历史上曾经出现多例 以与特性相关的 “假充值” 漏洞。
如我们主导披露的：
（1）USDT 虚假转账安全风险分析
（2）以太坊代币“假充值”漏洞细节披露及修复方案
及我们参与披露的：
XRP官方披露的假充值漏洞及相关分析

这些问题都不是链自己本身的漏洞，但是由于链自己本身的特性的原因，以及开发者对此类特性校验的不严谨从而导致了攻击的发生。这就是我们为什么会发布红色预警。这不是恐吓（FUD），更像是一种容易被人忽略的攻击手法。此类攻击手法之前在 EOS 上没有相同的攻击案例，但在我们公布相关的攻击手法后，根据我们联合 EOSPark 的数据分析系统发现，已经有十几个帐号开始分别对DApp、交易所以及钱包作出模仿攻击，这些帐号有：
bobukulabobu
cuancuan2323
testsuperdex
zhangjiayiba
justjiezhan1
wnze2qwdiyne
havls3k3iyge
ha11w4zzmpge
wkdoptxcjvdn
xmqukuailian
allosauruses
ccholr1ub2ku
walletthomas
fuckhakcer.x
.....等等其他帐号
其中不乏攻击成功的帐号。因此，我们继续发出后续预警，提醒相关项目方做好防范措施。除此之外，Twitter帐号 Whale Alert 也受到了此类攻击手法的影响。Whale Alert官方帐号在3月11日发布推文称帐号 fuckhacker.x 转账1万亿 EOS，随后被官方证明为虚假交易。可见此次攻击波及范围之广。应引起足够的重视。

修复方案

针对此类型的交易，相关项目方以及交易所及钱包需要对 EOS 转账的执行状态进行校验，确保交易执行状态为 “executed”，除此之外，在区块不可逆的情况下，也要做到以下几点防止其他类型的 “假充值” 攻击的发生

1、 判断 action 是否为 transfer
2、判断合约账号是否为 eosio.token 或其它 token 的官方合约

3、判断代币名称及精度
4、判断金额
5、判断 to 是否是自己平台的充币账号

后记 Q&A

Q：节点开启 read only 模式能防范此类攻击吗?
A：根据官方文档对read only模式的描述

Nodeos can be run in three modes:

1. **Speculative** In "speculative" mode you can see transactions and state changes up to the current head block plus changes in the pending block. The state changes in the pending block are also included in the chain state file.
2. **Head** In "head" mode you can see transactions and state changes up to the current head block. This node will produce a pending block, though the pending block is not included in the chain state file.
3. **Read only** In "read-only" mode you can see transactions and state changes up to the current head block. This node will NOT produce a pending block.

- The default mode is speculative.
- Chain state is a memory mapped file containing the state of the blockchain for each block. A record of chain state is kept for each block going back to the last irreversible block.
- The pending block is the block currently being built by each node, transactions are added to the pending block as they are received and processed. The pending block becomes the head block once it is written to the blockchain.

Nodes used for monitoring transactions should be run in:

- Speculative mode to see transactions as they arrive (Confirmed and unconfirmed)
- Read-Only mode to see transactions once they are recorded in the blockchain (Confirmed only)

For more information see Read Modes

节点开启 read only 模式后可以查询到线上记录并存在的确认的交易。而此类攻击手法是先发出 defer 交易，defer 交易是在链上保存的真实存在的交易，随后这笔交易才会转变成 hard_fail，所以开启 read only 模式并不能防御此类攻击手法。交易的状态是从 delay 转变成 hard_fail，并不是回滚，这是需要注意的地方。

Q：为什么我在其他的区块浏览器上，如 EOSPark 等不能查询到 hard_fail 交易?
A：现有的查询交易是通过 EOS 的历史插件 history plugin 来查询的，而根据 history plugin 的代码实现

```
    void on_applied_transaction( const transaction_trace_ptr& trace ) {
       if( !trace->receipt || (trace->receipt->status != transaction_receipt_header::executed &&
            trace->receipt->status != transaction_receipt_header::soft_fail) )
          return;
       for( const auto& atrace : trace->action_traces ) {
          on_action_trace( atrace );
       }
    }      Daniel Larimer, 11 months ago • Started work necessary to refactor history plugin
};
```

可以发现，除了executed 和 soft_fail 交易外其他交易都无法查询到。

Q：是否使用 history plugin 获取交易即可避免此类攻击?

A：无法保证，不同的节点 history plugin 插件的实现方式不一，不排除节点提供方自己修改 history plugin 实现，导致查询到的交易存在除 exectued 及 soft_fail 外的其他状态。

Q:交易所及钱包等项目方平台该如何配置他们自己的节点免受攻击?
A：使用默认的 history plugin 配置，除此之外检查 EOS 转账的交易状态，确保交易执行状态为 “executed”。同时，也需要判断以下几点防止其他类型的 “假充值”
1、 判断 action 是否为 transfer
2、判断合约账号是否为 eosio.token 或其它 token 的官方合约
3、判断代币名称及精度
4、判断金额
5、判断 to 是否是自己平台的充币账号
这些点都是曾在历史上出现过问题的点，我们认为 EOS 生态是一个很强大，灵活的全新的生态，从 EOS 主网启动以来，我们参与了许多的安全应急事件，开发者想在这个生态内安全的发展的确是一件充满挑战的事情。需要做好各方面的安全检查，才能确保资产不受损失。

结语

我们希望阅读此文的 EOS 生态中的用户不要模仿文中的攻击手法，我们无意对 EOS 社区造成恐慌。慢雾安全团队本着为 EOS 生态带来更多的安全感为宗旨，及时披露相关安全事件细节，目的是为了 EOS 生态中的更多成员获得安全保障。相关项目方应及时做好充提系统的安全保障，落实对应的风控策略，保障自身财产的安全。

相关参考

官方文档对nodeos三种模式的描述：https://developers.eos.io/eosio-nodeos/docs/read-modes
USDT 虚假转账安全风险分析：https://mp.weixin.qq.com/s/CtAKLNeoMOKDyUFaod4_hw
以太坊代币“假充值”漏洞细节披露及修复方案：https://mp.weixin.qq.com/s/3cMbE6p_4qCdVLa4FNA5-A
XRP官方披露的假充值漏洞及相关分析https://developers.ripple.com/partial-payments.html

致谢

EOSPark
IMEOS
jerry@EOSlive钱包