MDEX 交易合约疑似存在抽池漏洞及权限

注意：这是在 heco 链上，不是 eth 链

MDEX交易合约存在抽池漏洞及权限 ☆

发件人:

时　间: 2021年11月21日（星期日）下午8：06

收件人: developer <developer@mdex.com>

抄　送: team <team@slowmist.com>; market <market@lianantech.com>; support <support@noneage.com>

发送状态: 投递成功 [查看详情]

https://hecoinfo.com/tx/0x8e5b2e425f1dbed4289d7aa47a9e7ba01d0b779c0078ed6cafba803b2acf1429

MDEX的交易合约一直允许一个不知名的地址及合约调用TOKEN的交易对资金池抽离HT

地址如下：0xefea0b7c4506d84c1f650b3c058e35a5ba773540

合约如下：0x01aa4f3b56c99cf16049a16abce9798426509f16

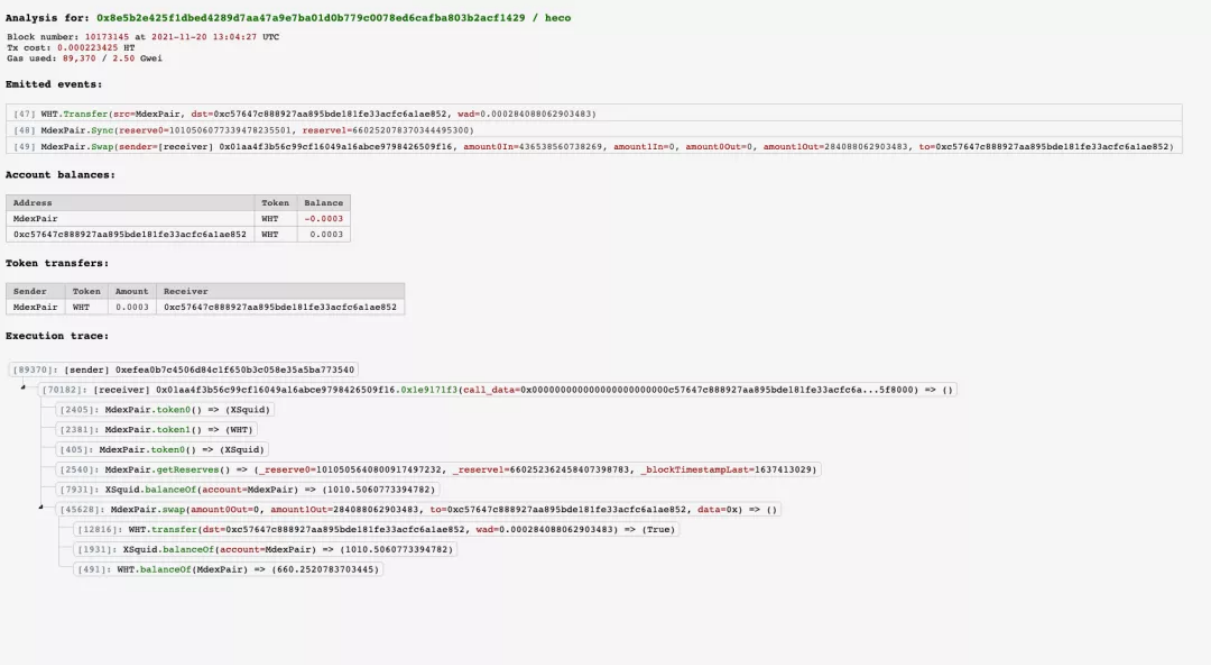
这个合约一直从各种交易对的池子一直抽取HT出来，额度非常小，但是很频繁。

严重怀疑 MDEX 的交易合约存在抽资金的权限，希望技术认真排查

漏洞类型

通缩映射型代币与项目合约不兼容

分析



1. 观察这笔交易,这是 XSquid 和 HT 的 Mdex Pair 池, Pair 合约通过 getReserves 获取到池子中_reserve0=1010.505640800917497232，在下一步通过 XSquid 合约 balanceOf 获取 pair 余额后结果为 1010.5060773394782
2. 定位到 XSquid 合约的 balanceOf

```
function balanceOf(address account) public view override returns (uint256) {
    if (_isExcluded[account]) return _tOwned[account];
    return tokenFromReflection(_rOwned[account]);
}
```

```
function tokenFromReflection(uint256 rAmount) public view returns(uint256) {
    require(rAmount <= _rTotal, "Amount must be less than total reflections");
    uint256 currentRate =  _getRate();
    return rAmount.div(currentRate);
}
```

```
function _getRate() private view returns(uint256) {
    (uint256 rSupply, uint256 tSupply) = _getCurrentSupply();
    return rSupply.div(tSupply);
}
```

```
function _getCurrentSupply() private view returns(uint256, uint256) {
    uint256 rSupply = _rTotal;
    uint256 tSupply = _tTotal;
    for (uint256 i = 0; i < _excluded.length; i++) {
        if (_rOwned[_excluded[i]] > rSupply || _tOwned[_excluded[i]] > tSupply) return (_rTotal, _tTotal);
        rSupply = rSupply.sub(_rOwned[_excluded[i]]);
        tSupply = tSupply.sub(_tOwned[_excluded[i]]);
    }
    if (rSupply < _rTotal.div(_tTotal)) return (_rTotal, _tTotal);
    return (rSupply, tSupply);
}
```

_tTotal 的总量是不会变化的，因此影响_getCurrentSupply 的输出结果是由_rTotal 决定的

```
702     uint256 private constant MAX = ~uint256(0);
703     uint256 private _tTotal = 1000000000 * 10**6 * 10**9;
704     uint256 private _rTotal = (MAX - (MAX % _tTotal));
705     uint256 private _tFeeTotal;
```

3. XSquid 是通缩映射型代币(通缩:代币阈值为 x,代币价格低于 x,则代币通缩；映射:x 个 A 代币可以置换 y 个 B 代币),每次转账时计算_rTotal 都会由_reflectFee 产生通缩使得_rTotal 值减小,而造成 currentRate 减少,而 rAmount.div(currentRate) == rAmount/currentRate 增大,最终造成所获取到的 balanceOf 大于 getReservers 获取到的值

```
function _reflectFee(uint256 rFee, uint256 tFee) private {
    _rTotal = _rTotal.sub(rFee);
    _tFeeTotal = _tFeeTotal.add(tFee);
}
```

4. 池子认为外部多打入了 XSquid,这时候攻击者调用 Mdex Pair 合约的 swap 函数根据上述计算的差额来抽取代币,或是调用 skim 函数直接转走代币. 因此就可以从池子里抽离这一小部分 HT.

5. 改进: 可以在每次转账最后通过调用 sync 函数强制准备金与余额同步更新