

区块链RPC攻击借“链”还魂，小蚁币持有者警惕盗币风险

作者：王凯（Kame Wang）、陈楠（Abnern Chen）@ 腾讯安全湛沪实验室

近日，腾讯安全湛沪实验室监测到著名区块链项目NEO（对应数字货币“小蚁币”，总市值1.8亿美元）存在远程盗币风险。用户在利用默认配置启动带有RPC功能的NEO网络节点并启用钱包时，其数字货币可能惨遭远程盗币。该攻击场景与今年年初被广泛披露的以太坊RPC攻击极为相似，严重危害了NEO用户的数字资产安全。

腾讯安全湛沪实验室对于该安全问题开展了紧急的研究分析，发现NEO节点确实存在着RPC接口暴露问题。该问题在[neo-cli客户端](#) v2.9.0版本之前广泛存在。此版本之后的节点在配置不当的情况下同样面临着一定的安全威胁。

因此，腾讯安全湛沪实验室特此发布此安全预警，提醒NEO节点维护者及小蚁币持有者警惕远程盗币攻击，并给出相应的安全防范建议以规避此类攻击带来的风险。

2. 重温以太坊RPC接口安全事件

在开始对于NEO节点RPC攻击开展研究分析之前，有必要对于以太坊RPC接口安全进行重温，以类比分析此次爆出的NEO RPC安全问题的形成原因和可能带来的安全危害。

以太坊是当前著名的区块链公链项目。以太坊节点的RPC接口以JSON-RPC的形式对外提供调用，其底层为Http协议。以太坊网络中RPC攻击是针对对于对外暴露了RPC接口的以太坊节点开展的。一般情况下，节点维护者可以利用节点的RPC接口控制节点的行为，如签署和发布交易信息。然而，以太坊各种版本的客户端实现有效的RPC接口调用者的身份验证机制。这为攻击者开展远程攻击提供了便利。

以太坊上最为典型的RPC攻击是远程盗币。要想成功实现攻击，需要：

- 获知节点维护钱包中的账户信息，攻击者可以通过调用eth_account等RPC接口获知。
- 在账户被解锁的状态下触发转账。要想实现转账，需用户使用正确密码解锁相应账户。账户解锁后，账户默认有300秒的时间窗口处于解锁状态。攻击者可以利用该时间窗抢先进行数字资产的窃取。

在以太坊网络中，已知的最早进行远程窃币的攻击发生于2016年2月14日。截至目前，该攻击者账户共窃取了4万5千余以太币，其市场价值超过5百万美金。仅此一个攻击者账户就证明了以太坊RPC攻击带来的严重的安全威胁。腾讯安全湛沪实验室于今年TSec安全会议上分享的名为《看好你的钱包！从攻击和防御角度分析以太坊RPC攻击》的安全研究报告中，利用蜜罐技术和脆弱节点扫描技术对于这一问题进行了深入的研究、分析。笔者认为，此研究手法同样适用于NEO节点上的RPC安全问题。

3. NEO区块链RPC接口安全问题

3.1 NEO节点的RPC接口是如何暴露的

NEO区块链的常用客户端之一neo-cli支持RPC接口，其实现形式同样采用的是JSON-RPC的方式。我们以存在安全问题的v2.9.0版本为例，在使用neo-cli --rpc启动节点时，所使用的默认RPC相关配置文件如下：

```
"RPC": {
  "Port": 10332,
  "sslCert": "",
  "sslCertPassword": ""
},
```

可以看出：

1. 该配置将SSL相关配置设置为空字符串，因此整个JSON-RPC接口底层基于Http而非Https实现，缺少了对于RPC调用者的身份验证；
2. 给定了RPC端口信息，但没有设置其将要绑定的IP地址。

利用该配置启动测试程序后，查看neo-cli开启的IP端口，可得到如下结果：

```
$ netstat -a -p
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
PID/Program name					
tcp	0	0	0.0.0.0:10332	0.0.0.0:*	LISTEN
3919/./neo-cli					
tcp	0	0	0.0.0.0:10333	0.0.0.0:*	LISTEN
3919/./neo-cli					

可以看出，neo-cli在缺省情况下将RPC接口直接绑定到了0.0.0.0地址，从而导致了RPC接口的对外暴露。实际上，在v2.9.0及其之前版本，并没有提供配置RPC接口绑定IP地址的方式，只要开启了RPC功能，就会被绑定到0.0.0.0地址；而在之后的版本中，开发者增加了新的RPC配置项`BindAddress`，其默认配置为127.0.0.1，这一配置能确保其接口安全性。但如果节点配置人员修改了该选项为0.0.0.0或其公网IP地址，也将使其面临着RPC攻击的威胁。

3.2 暴露RPC接口的NEO节点面临着怎样的安全威胁

neo-cli的诸多RPC接口均需要该节点所操作的钱包处于打开状态。通常情况下，节点本地操作者可以在neo-cli提供的本地命令接口中输入`open wallet`指令以打开相应钱包。

值得注意的是，目前neo-cli并未设置钱包自动关闭时间窗口。因此，用户的钱包在打开之后会长期处于开启状态，我们对于neo-cli提供的RPC接口进行了分析，筛选出了可能被攻击者实施攻击的相应接口：

方法	参数	说明
listaddress		列出当前钱包内的所有地址
getaccountstate		根据账户地址，查询账户资产信息
dumpprivkey		导出指定地址的私钥
sendrawtransaction		广播交易
sendfrom	<asset_id> [fee=0]	从指定地址，向指定地址转账
sendtoaddress	<asset_id> [fee=0]	向指定地址转账
sendmany	<outputs_array> [fee=0] [change_address]	批量转账命令
invoke	<script_hash>	使用给定的参数以散列值调用智能合约
invokefunction	<script_hash>	以指定的脚本散列值调用智能合约，传入操作及参数
invokescript	<script>	通过虚拟机运行脚本并返回结果

可以看出，攻击者依次调用listaddress和getaccountstate接口即可获知目标节点所维护的钱包及账户资产信息，而后可以采用的共计手段有：

1. 调用dumpprivkey，窃取受害者的私钥；
2. 调用sendXXX，窃取数字货币；
3. 调用invokeXXX，以受害者身份调用NEO上的智能合约。

3.3 攻击行为监测

在发现NEO RPC攻击存在后，腾讯安全湛沪实验室紧急部署了蜜罐系统对攻击者行为进行检测，并捕获到如下图所示的典型攻击行为。根据蜜罐日志我们可以看出，攻击者的攻击流程为：

1. 利用listaddress命令获取了我们的钱包地址；
2. 调用getaccountstate获取了钱包的asset信息；
3. 利用asset信息获取账户余额，确认有攻击价值；
4. 调用dumpprivkey，盗取账户私钥。
5. 当攻击者成功盗取用户的私钥之后，就可以攻击者名义签署交易，盗取货币。

```
[REQ] [2018-12-05 06:21:36] 82.157.191.47389 - [{"params": [], "jsonrpc": "2.0", "id": 0, "method": "listaddress"}]
[RESP] [2018-12-05 06:21:36] [{"jsonrpc": "2.0", "id": 0, "result": [{"address": "A6c...", "haskey": true, "label": null, "watchonly": false}]]
[REQ] [2018-12-05 06:21:37] 82.157.191.48843 - [{"params": ["A6c..."], "jsonrpc": "2.0", "id": 0, "method": "getaccountstate"}]
[RESP] [2018-12-05 06:21:37] [{"jsonrpc": "2.0", "id": 0, "result": {"version": 0, "script_hash": "0...", "frozen": false, "votes": [{"asset": "0x...", "value": "8"}]}]}]
[REQ] [2018-12-05 06:21:37] 82.157.191.50367 - [{"params": ["0..."], "jsonrpc": "2.0", "id": 0, "method": "getbalance"}, {"params": ["0..."], "jsonrpc": "2.0", "id": 1, "method": "getbalance"}, {"params": ["0..."], "jsonrpc": "2.0", "id": 2, "method": "getbalance"}, {"params": ["0..."], "jsonrpc": "2.0", "id": 3, "method": "getbalance"}]
[RESP] [2018-12-05 06:21:37] [{"jsonrpc": "2.0", "id": 0, "result": {"balance": "0", "confirmed": "0"}}, {"jsonrpc": "2.0", "id": 1, "result": {"balance": "0", "confirmed": "0"}}, {"jsonrpc": "2.0", "id": 2, "result": {"balance": "0", "confirmed": "0"}}, {"jsonrpc": "2.0", "id": 3, "result": {"balance": "0", "confirmed": "0"}}]
[REQ] [2018-12-05 06:21:38] 82.157.191.52013 - [{"params": ["A6c..."], "jsonrpc": "2.0", "id": 0, "method": "dumpprivkey"}]
[RESP] [2018-12-05 06:21:38] [{"jsonrpc": "2.0", "id": 0, "result": "K..."}]
```

3.4 脆弱节点扫描

在确认NEO网络中依然潜伏着RPC攻击者之后，腾讯安全湛沪实验室对于其网络规模及存在安全风险的节点进行了扫描。首先基于NEO节点P2P通信协议实现的网络节点探测接口发现了公网范围内的所有NEO节点。而后，腾讯安全湛沪实验室对这些节点的默认RPC端口（10332）进行了扫描，经测试发现有13.6%的节点暴露了其RPC接口，响应了我们发送的获取节点版本信息的RPC请求。在两天时间段内观察，发现其中有13.1%的节点会开启钱包，泄露钱包信息。



4. 安全防御建议

下面分别从NEO节点维护者和neo-cli开发者的角度提出一些安全建议，以提高其安全性能。

给NEO节点维护者的建议：

1. 升级到最高版本的neo-cli客户端程序；
2. 避免使用远程RPC功能，修改配置文件中BindAddress的地址为127.0.0.1；
3. 如有特殊需求，不得不使用远程RPC功能，应采取修改RPC端口号、启用基于Https的JSON-RPC接口、设置防火墙策略等方式保障节点安全。

给neo-cli开发者的建议：

1. 通知其社区成员及节点维护者尽快完成客户端更新；
2. 完全废弃基于Http的JSON-RPC功能，以Https为底层协议；
3. 修改代码逻辑，将钱包的“打开（解锁）-操作-关闭（上锁）”作为一个互斥的原子性事务进行处理，从而确保账户的安全敏感窗口期不被攻击者所利用。