Computer Security **Name:** Ameer Hamza
Assignment#5: Smart Contract Reentrancy Attack Lab

1. **Environment setup:** I picked the smaller containers in `output-small/` directory. I ran the docker commands, which is currently showing on left. Also, installed the `web3` library, shown on right.

Some intermediate output for running the containers.

```
Creating as160r-router0-10.160.0.254 ...
Creating as152r-router0-10.152.0.254 ...
Creating output-small_f1d53a66de3c35d8a921558f3b4bdbbd_1 ...
Creating output-small_cfee3a34e9c68ac1d16035a81a926786_1 ...
Creating as161h-host_0-10.161.0.71                       ...
Creating as3r-r104-10.104.0.3                            ...
Creating as161h-host_0-10.161.0.71                         ... done
Creating output-small_f1d53a66de3c35d8a921558f3b4bdbbd_1   ... done
Creating as4r-r102-10.102.0.4                              ...
Creating output-small_cfee3a34e9c68ac1d16035a81a926786_1   ... done
Creating as12r-r104-10.104.0.12                            ...
Creating as153h-Ethereum-POA-6-BootNode-Signer-10.153.0.71 ... done
Creating as150h-Ethereum-POA-0-BootNode-Signer-10.150.0.71 ...
Creating as12r-r101-10.101.0.12                            ... done
Creating as153h-Ethereum-POA-7-10.153.0.72                 ...
Creating as162h-host_1-10.162.0.72                         ...
Creating as2r-r102-10.102.0.2                              ... done
Creating as153r-router0-10.153.0.254                       ... done
Creating as4r-r100-10.100.0.4                              ... done
Creating as154h-Ethereum-POA-8-Signer-10.154.0.71          ... done
Creating as164h-host_0-10.164.0.71                         ... done
Creating as103rs-ix103-10.103.0.103                        ... done
Creating as102rs-ix102-10.102.0.102                        ... done
Creating as3r-r103-10.103.0.3                              ... done
Creating as150h-Ethereum-POA-1-10.150.0.72                 ... done
Creating as152h-Ethereum-POA-4-Signer-10.152.0.71          ... done
Creating as164h-host_1-10.164.0.72                         ... done
Creating as163h-host_0-10.163.0.71                         ... done
Creating as4r-r104-10.104.0.4                              ... done
Creating as164r-router0-10.164.0.254                       ... done
Creating as154h-Ethereum-POA-9-BootNode-10.154.0.72        ... done
Creating as151h-Ethereum-POA-3-BootNode-10.151.0.72        ... done
```

```
Creating as4r-r104-10.104.0.4                           ... done
Creating as164r-router0-10.164.0.254                    ... done
Creating as154h-Ethereum-POA-9-BootNode-10.154.0.72     ... done
Creating as151h-Ethereum-POA-3-BootNode-10.151.0.72     ... done
Creating as162h-host_0-10.162.0.71                      ... done
Creating as163h-host_1-10.163.0.72                      ... done
Creating as161h-host_1-10.161.0.72                      ... done
Creating as151r-router0-10.151.0.254                    ... done
Creating as160h-host_0-10.160.0.71                      ... done
Creating as160h-host_1-10.160.0.72                      ... done
Creating as100rs-ix100-10.100.0.100                     ... done
Creating as104rs-ix104-10.104.0.104                     ... done
Creating as154r-router0-10.154.0.254                    ... done
Creating as163r-router0-10.163.0.254                    ... done
Creating as101rs-ix101-10.101.0.101                     ... done
Creating as152h-Ethereum-POA-5-10.152.0.72              ... done
Creating as3r-r100-10.100.0.3                           ... done
Creating as2r-r101-10.101.0.2                           ... done
Creating as150r-router0-10.150.0.254                    ... done
Attaching to as161h-host_0-10.161.0.71, output-small_f1d53a66de3c35d8a921558f3b4bdbbd_1, as151h-Ethereum-POA-2-Signer-10.151.0.71, output-sma
ll_cfee3a34e9c68ac1d16035a81a926786_1, as153h-Ethereum-POA-6-BootNode-Signer-10.153.0.71, as154h-Ethereum-POA-8-Signer-10.154.0.71, as12r-r10
1-10.101.0.12, as150h-Ethereum-POA-1-10.150.0.72, as103rs-ix103-10.103.0.103, as164h-host_0-10.164.0.71, as102rs-ix102-10.102.0.102, as153h-E
thereum-POA-7-10.153.0.72, as150h-Ethereum-POA-0-BootNode-Signer-10.150.0.71, as162h-host_1-10.162.0.72, as163h-host_0-10.163.0.71, as12r-r10
4-10.104.0.12, as160r-router0-10.160.0.254, as164h-host_1-10.164.0.72, as3r-r104-10.104.0.3, as2r-r100-10.100.0.2, as152h-Ethereum-POA-4-Sign
er-10.152.0.71, as4r-r102-10.102.0.4, as152r-router0-10.152.0.254, as161r-router0-10.161.0.254, as162r-router0-10.162.0.254, as2r-r102-10.102
.0.2, as4r-r100-10.100.0.4, as154h-Ethereum-POA-9-BootNode-10.154.0.72, as153r-router0-10.153.0.254, as151h-Ethereum-POA-3-BootNode-10.151.0.
72, as163h-host_1-10.163.0.72, as160h-host_0-10.160.0.71, as162h-host_0-10.162.0.71, as3r-r103-10.103.0.3, as164r-router0-10.164.0.254, as161
h-host_1-10.161.0.72, as104rs-ix104-10.104.0.104, as100rs-ix100-10.100.0.100, as4r-r104-10.104.0.4, as160h-host_1-10.160.0.72, as152h-Ethereu
m-POA-5-10.152.0.72, as101rs-ix101-10.101.0.101, as3r-r100-10.100.0.3, as163r-router0-10.163.0.254, as151r-router0-10.151.0.254, as154r-route
r0-10.154.0.254, as150r-router0-10.150.0.254, as2r-r101-10.101.0.2
```

2. **Task 1: Getting Familiar with the Victim Smart Contract**

(a) Compiled the `ReentrancyVictim.sol` contract. The output shows the `bin` and `abi` files generated.

```
[04/06/23]seed@VM:~/.../contract$ solc-0.6.8 --overwrite --abi --bin -
o . ReentrancyVictim.sol
Compiler run successful. Artifact(s) can be found in directory ..
[04/06/23]seed@VM:~/.../contract$ ls ReentrancyVictim.*
ReentrancyVictim.abi  ReentrancyVictim.bin  ReentrancyVictim.sol
[04/06/23]seed@VM:~/.../contract$
```

(b) deploying the victim contract:

```
[04/06/23]seed@VM:~/.../victim$ python3 deploy_victim_contract.py
Sending tx ...
---------Deploying Contract ----------------
... Waiting for block
Transaction Hash: 0x2055abeeb02a58498f02a29ca8936ca86980f6a81c8a36da69be8a373bd40037
Transaction Receipt: AttributeDict({'blockHash': HexBytes('0x561e053c093574a480184e3206135f4eb979f5900c7b1c13d9ed1ada8e26cfac'), 'blockNumber
': 1509, 'contractAddress': '0xc9F7DC324a9F1ae3778010F2ef45787419915064', 'cumulativeGasUsed': 282261, 'effectiveGasPrice': 1000000007, 'from
': '0xA403f63AD02a557D5DDCBD5F5af9A7627C591034', 'gasUsed': 282261, 'logs': [], 'logsBloom': HexBytes('0x00000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000'), 'status': 1, 'to': None, 'transactionHash': HexBytes('0x2055abeeb02a58498f02a29ca893
6ca86980f6a81c8a36da69be8a373bd40037'), 'transactionIndex': 0, 'type': 2})
Victim contract: 0xc9F7DC324a9F1ae3778010F2ef45787419915064
[04/06/23]seed@VM:~/.../victim$ cat contract_address_victim.txt
0xc9F7DC324a9F1ae3778010F2ef45787419915064[04/06/23]seed@VM:~/.../victim$
```

(c) I ended up depositing 10 ether first, and then 30 ether. Hence, the screenshot shows both transactions. The total balance is 40 ether.

```
[04/06/23]seed@VM:~/.../victim$ python3 fund_victim_contract.py
Transaction sent, waiting for the block ...
Transaction Receipt: AttributeDict({'blockHash': HexBytes('0xf580cb0acbdfd556bad61748643514e80c51d264ef877e88b28a8b79cabf84a5'), 'blockNumber
': 1540, 'contractAddress': None, 'cumulativeGasUsed': 65581, 'effectiveGasPrice': 1000000007, 'from': '0xA403f63AD02a557D5DDCBD5F5af9A7627C5
91034', 'gasUsed': 65581, 'logs': [], 'logsBloom': HexBytes('0x00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000'), 'status': 1, 'to': '0xc9F7DC324a9F1ae3778010F2ef45787419915064', 'transactionHash': HexBytes('0xa87c59c2e6de79b70812c66c730b64
85cdc674b44874d2b55da6c5c4fa7fe0ae'), 'transactionIndex': 0, 'type': 2})
------------------------------------------------------------
== My balance inside the contract:
   0xA403f63AD02a557D5DDCBD5F5af9A7627C591034: 10000000000000000000
== Smart Contract total balance:
   0xc9F7DC324a9F1ae3778010F2ef45787419915064: 10000000000000000000
------------------------------------------------------------
[04/06/23]seed@VM:~/.../victim$ vim fund_victim_contract.py
[04/06/23]seed@VM:~/.../victim$ python3 fund_victim_contract.py
Transaction sent, waiting for the block ...
Transaction Receipt: AttributeDict({'blockHash': HexBytes('0xa2050452e2b08f46bad04e95bceb899a49fa8ad3eb50722c185b02b3da0e5b64'), 'blockNumber
': 1543, 'contractAddress': None, 'cumulativeGasUsed': 31381, 'effectiveGasPrice': 1000000007, 'from': '0xA403f63AD02a557D5DDCBD5F5af9A7627C5
91034', 'gasUsed': 31381, 'logs': [], 'logsBloom': HexBytes('0x00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000'), 'status': 1, 'to': '0xc9F7DC324a9F1ae3778010F2ef45787419915064', 'transactionHash': HexBytes('0xfd47ee6dd89220132053a15c3f02ba
a8ba96bf80ae07ba0b4eb62a35316f87d6'), 'transactionIndex': 0, 'type': 2})
------------------------------------------------------------
== My balance inside the contract:
   0xA403f63AD02a557D5DDCBD5F5af9A7627C591034: 40000000000000000000
== Smart Contract total balance:
   0xc9F7DC324a9F1ae3778010F2ef45787419915064: 40000000000000000000
```

(d) Since I deposited 40 ether, I will now withdraw 15 ether, instead of 5 ether (as instructed). The final amount is 25 ether. This is to make sure the assignment tasks are consistent (if this amount is needed in the future).

```
[04/06/23]seed@VM:~/.../victim$ python3 withdraw_from_victim_contract.py
Transaction sent, waiting for the block ...
Transaction Receipt: AttributeDict({'blockHash': HexBytes('0x1ce215c65f05e6245846d8305fd6642e405d7bb629dc1262955ddface0b42b06'), 'blockNumber
': 1568, 'contractAddress': None, 'cumulativeGasUsed': 38799, 'effectiveGasPrice': 1000000007, 'from': '0xA403f63AD02a557D5DDCBD5F5af9A7627C5
91034', 'gasUsed': 38799, 'logs': [], 'logsBloom': HexBytes('0x00000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000'), 'status': 1, 'to': '0xc9F7DC324a9F1ae3778010F2ef45787419915064', 'transactionHash': HexBytes('0xe3c767d4eb018b8617a66db7bd0d0e
53732a06eb9ac5944b0a1300aa62262194'), 'transactionIndex': 0, 'type': 2})
------------------------------------------------------------
== My balance inside the contract:
   0xA403f63AD02a557D5DDCBD5F5af9A7627C591034: 25000000000000000000
== Smart Contract total balance:
   0xc9F7DC324a9F1ae3778010F2ef45787419915064: 25000000000000000000
------------------------------------------------------------
```

3. **Task 2: The Attacking Contract**

   The attack contract has been deployed:

```
[04/06/23]seed@VM:~/.../attacker$ python3 deploy_attack_contract.py
---------Deploying Contract ----------------
... Waiting for block
Transaction Hash: 0xe112fb3c37705105261565570585 6a69a31b0dfe75363cf7f075aa741da37a83
Transaction Receipt: AttributeDict({'blockHash': HexBytes('0x7c9142fd48c5340cfbc04fb3b854b324cf3f1134ec3cba28f7d292e5d064e91e'), 'blockNumber
': 2269, 'contractAddress': '0x758a1930B1a2350F446f81f39E4D2E8e010227A2', 'cumulativeGasUsed': 356807, 'effectiveGasPrice': 1000000007, 'from
': '0x9105A373ce1d01B517aA54205A5E4c70FA9f34Fe', 'gasUsed': 356807, 'logs': [], 'logsBloom': HexBytes('0x000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000'), 'status': 1, 'to': None, 'transactionHash': HexBytes('0xe112fb3c37705105261565570585
6a69a31b0dfe75363cf7f075aa741da37a83'), 'transactionIndex': 0, 'type': 2})
Attack contract: 0x758a1930B1a2350F446f81f39E4D2E8e010227A2
```

4. **Task 3: Launching the Reentrancy Attack**

   The attack is launched.

```
[04/06/23]seed@VM:~/.../attacker$ python3 launch_attack.py
Transaction sent, waiting for block ...
Transaction Receipt: AttributeDict({'blockHash': HexBytes('0xf62c99b61e155a2e8e8fc42c72689d46824aaf94b96492e517483ba32a8370bd'), 'blockNumber
': 2292, 'contractAddress': None, 'cumulativeGasUsed': 282944, 'effectiveGasPrice': 1000000007, 'from': '0x9105A373ce1d01B517aA54205A5E4c70FA
9f34Fe', 'gasUsed': 282944, 'logs': [], 'logsBloom': HexBytes('0x0000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000'), 'status': 1, 'to': '0x758a1930B1a2350F446f81f39E4D2E8e010227A2', 'transactionHash': HexBytes('0xb713eed8733d7b9593f87672c75f
0931914e2048cf5e41edbeae62655f56db27'), 'transactionIndex': 0, 'type': 2})
```

   We can see that the victim contract has 0 balance (it had 25), while the attacker contract has 26 balance (25 stolen from the victim contract, 1 was deposited while calling the attack() method).

```
[04/06/23]seed@VM:~/.../attacker$ python3 get_balance.py
----------------------------------------------------------
*** This client program connects to 10.151.0.71:8545
*** The following are the accounts on this Ethereum node
0x8c400205fDb103431F6aC7409655ad3cf8f6d007: 32000070180000000000
0x9105A373ce1d01B517aA54205A5E4c70FA9f34Fe: 54999999999999999999989993602489955217 43
----------------------------------------------------------
  Victim: 0xc9F7DC324a9F1ae3778010F2ef45787419915064: 0
Attacker: 0x758a1930B1a2350F446f81f39E4D2E8e010227A2: 26000000000000000000
```

   I tried sending money to another account, specifically to web3.eth.accounts[2] from the sender account web3.eth.accounts[1]. However, I get the following error:

```
[04/12/23]seed@VM:~/.../attacker$ python3 cashout.py
Traceback (most recent call last):
  File "cashout.py", line 17, in <module>
    recipient_acct = Web3.to_checksum_address(web3.eth.accounts[2])
IndexError: list index out of range
```

   I can guess that there are only two accounts at the given node, so I end up getting an index error that there is no account. I suspect that no such account was created when creating the emulator, hence it seems to me out of scope of the assignment to try to add another account and recompile everything. I also tried some other IP addresses to see if another node is available with an account to send to, but I kept getting errors like `web3.exceptions.ContractLogicError: execution reverted`. I believe I can skip this part, as major part where we try to steal money is done.

5. **Task 4: Countermeasures**

I made the required countermeasure change and then rerun the attack. Here are some initial steps to deploy and fund the victim contract:

```
[04/12/23]seed@VM:~/.../victim$ python3 deploy_victim_contract.py
Sending tx ...
---------Deploying Contract ----------------
... Waiting for block
Transaction Hash: 0xb3abeae57734732dc6bd100956a82bfca204254ca1039ebb61dc13a7f00687be
Transaction Receipt: AttributeDict({'blockHash': HexBytes('0x38059d0b0a6e4034dfa307f204fd2c20e41515b9b53ebb0d0db16f0b411edb93'), 'blockNumber
': 2258, 'contractAddress': '0xbA977a3dF7c3E0A926Fa46EA749f3C182147f4F8', 'cumulativeGasUsed': 282249, 'effectiveGasPrice': 1000000007, 'from
': '0xA403f63AD02a557D5DDCBD5F5af9A7627C591034', 'gasUsed': 282249, 'logs': [], 'logsBloom': HexBytes('0x00000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000'), 'status': 1, 'to': None, 'transactionHash': HexBytes('0xb3abeae57734732dc6bd100956a8
2bfca204254ca1039ebb61dc13a7f00687be'), 'transactionIndex': 0, 'type': 2})
Victim contract: 0xbA977a3dF7c3E0A926Fa46EA749f3C182147f4F8
[04/12/23]seed@VM:~/.../victim$ vim fund_victim_contract.py
[04/12/23]seed@VM:~/.../victim$ python3 fund_victim_contract.py
Transaction sent, waiting for the block ...
Transaction Receipt: AttributeDict({'blockHash': HexBytes('0xd7a1553510062685d52e6169a6eee5dd590c505e11147e6014cb610614817aba'), 'blockNumber
': 2262, 'contractAddress': None, 'cumulativeGasUsed': 65581, 'effectiveGasPrice': 1000000007, 'from': '0xA403f63AD02a557D5DDCBD5F5af9A7627C5
91034', 'gasUsed': 65581, 'logs': [], 'logsBloom': HexBytes('0x00000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000'), 'status': 1, 'to': '0xbA977a3dF7c3E0A926Fa46EA749f3C182147f4F8', 'transactionHash': HexBytes('0xd1b45f45d6ab9ef04ca89169bc2313
32ddba7c6e1d386447c90047ee992f8ec1'), 'transactionIndex': 0, 'type': 2})
------------------------------------------------------------
== My balance inside the contract:
   0xA403f63AD02a557D5DDCBD5F5af9A7627C591034: 30000000000000000000
== Smart Contract total balance:
   0xbA977a3dF7c3E0A926Fa46EA749f3C182147f4F8: 30000000000000000000
------------------------------------------------------------
[04/12/23]seed@VM:~/.../victim$
```

then, I deployed the attack contract and tried to launch the attack. However, it was unsuccessful. Part of the error trace is shown below;

```
[04/12/23]seed@VM:~/.../attacker$ python3 deploy_attack_contract.py
---------Deploying Contract ----------------
... Waiting for block
Transaction Hash: 0xcfb28353c7b2ed82ac8e0240c4f90e6254076c401715ddb9aa44b8b21aa8bb81
Transaction Receipt: AttributeDict({'blockHash': HexBytes('0x1fb95585e8573f8396c8edbf1f10ebe4a3dcbd2dc5b81a81e0d5e630b2d3a79d'), 'blockNumber
': 2402, 'contractAddress': '0x508b995749FeDc4248e16fCa1121D55DF992B537', 'cumulativeGasUsed': 356807, 'effectiveGasPrice': 1000000007, 'from
': '0x9105A373ce1d01B517aA54205A5E4c70FA9f34Fe', 'gasUsed': 356807, 'logs': [], 'logsBloom': HexBytes('0x00000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000'), 'status': 1, 'to': None, 'transactionHash': HexBytes('0xcfb28353c7b2ed82ac8e0240c4f9
0e6254076c401715ddb9aa44b8b21aa8bb81'), 'transactionIndex': 0, 'type': 2})
Attack contract: 0x508b995749FeDc4248e16fCa1121D55DF992B537
[04/12/23]seed@VM:~/.../attacker$ vim launch_attack.py
[04/12/23]seed@VM:~/.../attacker$ python3 launch_attack.py
Traceback (most recent call last):
  File "launch_attack.py", line 18, in <module>
    tx_hash  = contract.functions.attack().transact({
  File "/home/seed/.local/lib/python3.8/site-packages/web3/contract/contract.py", line 479, in transact
    return transact_with_contract_function(
  File "/home/seed/.local/lib/python3.8/site-packages/web3/contract/utils.py", line 172, in transact_with_contract_function
    txn_hash = w3.eth.send_transaction(transact_transaction)
  File "/home/seed/.local/lib/python3.8/site-packages/web3/eth/eth.py", line 362, in send_transaction
    return self._send_transaction(transaction)
```

The last few lines of the trace might be more helpful. This error message "Failed to send Ether!" is printed in the withdraw method in the victim smart contract when `msg.sender.call` return `false` in the `sent` variable.

```
  File "/home/seed/.local/lib/python3.8/site-packages/web3/eth/eth.py", line 292, in estimate_gas
    return self._estimate_gas(transaction, block_identifier)
  File "/home/seed/.local/lib/python3.8/site-packages/web3/module.py", line 68, in caller
    result = w3.manager.request_blocking(
  File "/home/seed/.local/lib/python3.8/site-packages/web3/manager.py", line 232, in request_blocking
    return self.formatted_response(
  File "/home/seed/.local/lib/python3.8/site-packages/web3/manager.py", line 197, in formatted_response
    apply_error_formatters(error_formatters, response)
  File "/home/seed/.local/lib/python3.8/site-packages/web3/manager.py", line 73, in apply_error_formatters
    formatted_resp = pipe(response, error_formatters)
  File "cytoolz/functoolz.pyx", line 666, in cytoolz.functoolz.pipe
  File "cytoolz/functoolz.pyx", line 641, in cytoolz.functoolz.c_pipe
  File "/home/seed/.local/lib/python3.8/site-packages/web3/_utils/method_formatters.py", line 762, in raise_contract_logic_error_on_revert
    raise ContractLogicError(response["error"]["message"])
web3.exceptions.ContractLogicError: execution reverted: Failed to send Ether!
```