

Forum: [Hack Forums](#)

Thread: [Polygonscan Token Hijacking Concept "Pay The Gas"](#)

Clumsy - July 17, 2024, 12:32 pm

Python Script for Automated Wallet Connection and Contract Interaction on Polygonscan

Prerequisites

Install Required Packages:

Code pip install selenium

Download WebDriver:

Ensure ChromeDriver is in your system PATH.

MetaMask Extension:

Download and set up MetaMask in a Chrome profile.

Python Script

Code [python]

```
import os
```

import time

```
import logging
```

```
from selenium import webdriver
```

```
from selenium.webdriver.common.by import By
```

```
from selenium.webdriver.support.ui import WebDriverWait
```

```
from selenium.webdriver.support import expected_conditions as EC
```

```
# Set up logging
```

```
logging.basicConfig(level=logging.INFO)
```

```
logger = logging.getLogger(__name__)
```

```
# Function to set up the Selenium WebDriver with MetaMask extension
```

```
def setup_selenium_with_metamask(metamask_extension_path, chrome_profile_path):
```

```
options = webdriver.ChromeOptions()
```

```
options.add_extension(metamask_extension_path)
```

```
options.add_argument(f"user-data-dir={chrome_profile_path}")
```

```
driver = webdriver.Chrome(options=options)
```

```
return driver
```

Function to automate wallet connection using Selenium

```
def automate_wallet_connection(driver):
```

try:

[illegible]

```
#writeContract")
```

```
# Wait for the "Connect to Web3" button to be clickable and click it
connect_button = WebDriverWait(driver, 20).until(
    EC.element_to_be_clickable((By.XPATH, "//button[contains(text(), 'Connect to Web3')]"))
)
connect_button.click()
```

```
# Wait for the MetaMask option to be clickable and click it
metamask_option = WebDriverWait(driver, 20).until(
    EC.element_to_be_clickable((By.XPATH, "//button[contains(text(), 'MetaMask')]"))
)
metamask_option.click()
```

```
# Switch to the MetaMask extension popup
driver.switch_to.window(driver.window_handles[1])
```

```
# Confirm connection in MetaMask (this might need to be adjusted based on MetaMask's UI changes)
```

```
confirm_button = WebDriverWait(driver, 20).until(
    EC.element_to_be_clickable((By.XPATH, "//button[contains(text(), 'Next')]"))
)
confirm_button.click()
```

```
connect_button = WebDriverWait(driver, 20).until(
    EC.element_to_be_clickable((By.XPATH, "//button[contains(text(), 'Connect')]"))
)
connect_button.click()
```

```
# Switch back to the original window
driver.switch_to.window(driver.window_handles[0])
```

```
logger.info("Wallet connected successfully")
except Exception as e:
    logger.error(f"An error occurred during wallet connection: {e}")
```

```
# Main function
```

```
def main():
```

```
    metamask_extension_path = os.getenv('METAMASK_EXTENSION_PATH')
    chrome_profile_path = os.getenv('CHROME_PROFILE_PATH')
```

```
    if not all([metamask_extension_path, chrome_profile_path]):
        logger.error("Environment variables METAMASK_EXTENSION_PATH and CHROME_PROFILE_PATH are not set.")
    return
```

```
driver = setup_selenium_with_metamask(metamask_extension_path, chrome_profile_path)
```

```
try:
```

```
    # Automate wallet connection
    automate_wallet_connection(driver)
```

```
finally:
```

```
    driver.quit()
```

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

[/python]

Setup Instructions

Set Environment Variables:

Set METAMASK_EXTENSION_PATH and CHROME_PROFILE_PATH before running the script.

Run the Script:

Code python automated_withdrawal.py

Notes

MetaMask Handling:

Automating MetaMask interactions may require additional setup and handling based on MetaMask UI changes.

Environment Variables:

Ensure all environment variables are correctly set to avoid interruptions in script execution.

Web Driver:

Ensure the correct web driver (e.g., ChromeDriver) is accessible in your system PATH.

This script automates wallet connection and contract interaction on Polygonscan without exposing the private key, suitable for secure deployment.