```python
from selenium.webdriver.remote.webelement import WebElement
from selenium.webdriver.edge.webdriver import WebDriver
from selenium import webdriver
from selenium.webdriver.edge.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as ec
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.action_chains import ActionChains
from time import sleep

...


def create_driver():
    service = Service(WEBDRIVER_PATH)
    options = webdriver.EdgeOptions()
    options.add_argument(f'user-data-dir={USER_DATA_DIR}')
    options.headless = True
    driver_ = webdriver.Edge(service=service, options=options)
    driver_.implicitly_wait(10)
    return driver_


def load_last_posts() -> dict:
    r = dict()
    with open("last_posts_id.txt", "r") as f:
        for line in f:
            print(repr(line))
            try:
                g, p = line.split(':')
                r[g] = p.strip()
            except ValueError:
                pass
    return r


def dump_last_posts(r: dict):
    with open("last_posts_id.txt", "w") as f:
        for g_, p_ in r.items():
```

```python
def dump_last_posts(r: dict):
    with open("last_posts_id.txt", "w") as f:
        for g_, p_ in r.items():
            line = ':'.join((g_, p_))
            f.write(line + '\n')


def send_posts_from_group(driver_: WebDriver, group_: dict, last_post: str = ""):
    driver_.get(group_["url"])
    posts = driver_.find_elements(By.XPATH,
                                  '//div[@class="_post post page_block all own post_signed post--withPostBottomAction'
                                  '  post--with-likes deep_active Post--redesign"]')
    posts.extend(driver_.find_elements(By.XPATH,
                                       '//div[@class="_post post page_block all own post--withPostBottomAction'
                                       '  post--with-likes deep_active Post--redesign"]'))
    print("Длина posts: ", len(posts))
    print("Я нашел следующие посты, Уолтер:")
    for post in posts:
        print(post.get_attribute('id'))
    indx = 0
    while indx < len(posts):
        if posts[indx].get_attribute("id") == last_post:
            posts = posts[:indx]
            break
        else:
            indx += 1
    posts = posts[::-1]
    if not posts:
        print('Новых постов нет. Долбаные мексы')
        return last_post
    else:
        print('Посты для отправки, Уолтер: ')
        for post in posts:
            print(post.get_attribute('id'))
        for post in posts:
            print("Отправка", post.get_attribute('id'))
            send_post(driver_, post, group_["message"])
            sleep(2)
        return posts[-1].get_attribute('id')
```

```python
def send_post(driver_: WebDriver, post_: WebElement, message_: str):
    hernia = WebDriverWait(driver_, 10).until(ec.presence_of_element_located((By.XPATH,
                        f'//*[@id="{post_.get_attribute("id")}"]/div/div[2]/div/div[2]/div/div[1]/div[3]')))
    driver_.find_element(By.TAG_NAME, 'body').send_keys(Keys.CONTROL + Keys.HOME)
    action = ActionChains(driver_)
    action.scroll_to_element(hernia)
    action.perform()
    sleep(1)
    hernia.click()
    form = WebDriverWait(driver_, 10).until(ec.presence_of_element_located((By.CSS_SELECTOR, "form[class=\"like_share_wrap\"]")))
    imp = form.find_element(By.ID, 'like_im_inp')
    imp.send_keys(BOYS_CHAT_NAME)
    sleep(1)
    imp.send_keys(Keys.ENTER)
    driver_.find_element(By.ID, 'like_share_text').send_keys(message_)
    WebDriverWait(driver_, 10).until(ec.element_to_be_clickable((By.ID, "like_share_send"))).click()
    print('Успешно отправлен', post_.get_attribute("id"))
    WebDriverWait(driver_, 10).until(ec.invisibility_of_element_located((By.CSS_SELECTOR, "form[class=\"like_share_wrap\"]")))
    driver_.switch_to.parent_frame()


def main():
    driver = create_driver()
    last_posts = load_last_posts()
    for group in GROUPS:
        print(f"Исследую {group['name']}, Уолтер")
        new_last_post = send_posts_from_group(driver, group, last_posts.get(group["name"], ""))
        last_posts[group["name"]] = new_last_post
    dump_last_posts(last_posts)
    driver.quit()
```

```python
def main():
    driver = create_driver()
    last_posts = load_last_posts()
    for group in GROUPS:
        print(f"Исследую {group['name']}, Уолтер")
        new_last_post = send_posts_from_group(driver, group, last_posts.get(group["name"], ""))
        last_posts[group["name"]] = new_last_post
    dump_last_posts(last_posts)
    driver.quit()


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