개요

이번 화이트햇스쿨 1기 1차 CTF에 나온 give me present라는 웹해킹 문제에 대한 Writeup입니다.

분석한 내용

우선 해당 문제에 소스코드는 다음과같다.

```
#!/usr/bin/python3
import os
from flask import Flask, request
from flask import make_response, render_template
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
import urllib

app = Flask(__name__)
app.secret_key = os.urandom(32)

try:
    FLAG = open('./flag.txt', 'r').read()
except:
    FLAG = '[**FLAG**]'


def read_url(url):
    try:
        service = Service(executable_path="/chromedriver")
        options = webdriver.ChromeOptions()
        for _ in [
            "headless",
            "window-size=1920x1080",
            "disable-gpu",
            "no-sandbox",
            "disable-dev-shm-usage",
        ]:
            options.add_argument(_)
        driver = webdriver.Chrome(service=service, options=options)
        driver.implicitly_wait(3)
        driver.set_page_load_timeout(3)
        driver.get("http://127.0.0.1:8000/")
        driver.get(url)
        driver.get("http://127.0.0.1:8000/check-present")
        driver.get("http://127.0.0.1:8000/memo")
    except:
        driver.quit()
        return False
    driver.quit()
    return True
```

```
def admin_present(sender, present):
    url = f"http://127.0.0.1:8000/present?sender=
{urllib.parse.quote(sender)}&present={urllib.parse.quote(present)}"
    return read_url(url)


@app.route("/")
def index():
    return render_template("index.html")


@app.route('/present', methods=['GET'])
def present():
    if request.method == 'GET':
        sender = request.args.get('sender', 'Sender-Name')
        present = request.args.get('present', '')

        if present == '':
            message = "No present now..."
        else:
            message = f"{sender} gave you: {present}"

        resp = make_response(render_template("present.html", message=message))
        resp.headers.set(sender, present)

        return resp
    else:
        return '<script>alert("no!");history.go(-1);</script>'


@app.route('/give-present', methods=['GET', 'POST'])
def give_present():
    if request.method == 'GET':
        return render_template("give-present.html")
    elif request.method == 'POST':
        sender = request.form.get("sender")
        present = request.form.get("present")
        if not admin_present(sender, present):
            return '<script>alert("wrong??");history.go(-1);</script>'
        return '<script>alert("good");history.go(-1);</script>'


@app.route('/check-present', methods=['GET'])
def check_present():
    if request.method == 'GET':
        present = request.cookies.get('present', '')
        resp = make_response(render_template("check-present.html"))
        print(request.environ['REMOTE_ADDR'])
        if present == 'money' and request.environ['REMOTE_ADDR'] == '127.0.0.1':
            resp.set_cookie("flag", FLAG)
        return resp
    else:
        return render_template("check-present.html")
```

```
memo_text = ""


@app.route("/memo")
def memo():
    global memo_text
    text = request.cookies.get('flag', '')
    memo_text += text + "\n"
    return render_template("memo.html", memo=memo_text)


app.run(host='0.0.0.0', port=8000)
```

부분부분 분석을 해보자면 우선 처음 FLAG를 설정하는 부분이 있는데, 해당 부분은 기본적 설정방식이고 자주 나오니 이 Write up에서만 설명하고 다른 Write up에서는 생략하겠다.

```
try:
    FLAG = open('./flag.txt', 'r').read()
except:
    FLAG = '[**FLAG**]'
```

try부분의 코드는 이 파일의 현재 위치에 존재하는 `flag.txt`라는 파일을 읽기모드로 문자열들을 읽어와 FLAG라는 변수에 저장하고, 만약 해당 디렉토리에 `flag.txt`파일이 없거나 다른 오류가 날 경우 `[**FLAG**]`라는 문자열을 FLAG라는 변수에 저장해 flag문자를 대체하는 코드이다. 즉 이부분의 코드는 실제 flag의 문자열을 문제파일에 넣어 줄 수 없으니 문제파일을 받아 개인 local에서 서버를 실행시킬경우 flag문자를 찾지못하는 오류를 방지해주는 코드이다.

```
def read_url(url):
    try:
        service = Service(executable_path="/chromedriver")
        options = webdriver.ChromeOptions()
        for _ in [
            "headless",
            "window-size=1920x1080",
            "disable-gpu",
            "no-sandbox",
            "disable-dev-shm-usage",
        ]:
            options.add_argument(_)
        driver = webdriver.Chrome(service=service, options=options)
        driver.implicitly_wait(3)
        driver.set_page_load_timeout(3)
        driver.get("http://127.0.0.1:8000/")
        driver.get(url)
        driver.get("http://127.0.0.1:8000/check-present")
        driver.get("http://127.0.0.1:8000/memo")
    except:
        driver.quit()
```

```
            return False
        driver.quit()
        return True
```

이 코드에서는 webdriver를 통해 서버에서 가상?의 브라우저를 통해 일련의 행동을 시키는것이다. 문제를 풀때 필요치 않은 부분을 제외하고 설명하자면, 처음 `127.0.0.1:8000`으로 현재 자신의 서버에 루프백 ip를 통해 get요청 하고, 파라미터로받은 url의 주소로 get요청을 하고, `/cehck-present`라는 주소로 get요청을 하고, `/memo`라는 주소로 get요청을 하는 과정을 하고 종료를 하는 함수이다.

```
@app.route('/present', methods=['GET'])
def present():
    if request.method == 'GET':
        sender = request.args.get('sender', 'Sender-Name')
        present = request.args.get('present', '')

        if present == '':
            message = "No present now..."
        else:
            message = f"{sender} gave you: {present}"

        resp = make_response(render_template("present.html", message=message))
        resp.headers.set(sender, present)

        return resp
    else:
        return '<script>alert("no!");history.go(-1);</script>'
```

해당 코드는 `/present`의 url로 접속했을 때 의 코드이다. 만약 `GET`요청을 받을 경우 sender와 present에 각각 url 파라미터로 전달받은 인자를 저장한다. 근데 present파라미터로 전달받은 값이 없다면 페이지가 다시 렌더링되고 "No presnet now..."을 출력한다. present값이 존재할경우 페이지가 다시 렌더링되고 sender와 present의 값을 넣은 "{sender} gave you: {present}"를 출력한다. 그리고 응답을 줄 때 header를 `{sender}: {present}`의 형식으로 설정을 해준다.

```
@app.route('/give-present', methods=['GET', 'POST'])
def give_present():
    if request.method == 'GET':
        return render_template("give-present.html")
    elif request.method == 'POST':
        sender = request.form.get("sender")
        present = request.form.get("present")
        if not admin_present(sender, present):
            return '<script>alert("wrong??");history.go(-1);</script>'
        return '<script>alert("good");history.go(-1);</script>'
```

해당코드는 `/give-present`에 접속할 경우 실행이된다. url은 GET요청방식과 POST요청방식모두 허가하고있다. 만약 GET요청을 받을 경우 그냥 html을 렌더링해주고 POST요청을 받았을 경우 sender와 presend에 파라미

터로받은 값을 각각 저장한다. 이후 admin_present(sender, present)실행하고 return값이 존재할 경우 good을 띄우고 존재하지 않을경우 wrong을 띄운다.

```
@app.route('/check-present', methods=['GET'])
def check_present():
    if request.method == 'GET':
        present = request.cookies.get('present', '')
        resp = make_response(render_template("check-present.html"))
        print(request.environ['REMOTE_ADDR'])
        if present == 'money' and request.environ['REMOTE_ADDR'] == '127.0.0.1':
            resp.set_cookie("flag", FLAG)
        return resp
    else:
        return render_template("check-present.html")
```

해당 코드는 `/check-present`url에 접속할 경우 실행이된다. 해당 코드는 GET요청만 허용을 하고있다. GET요청을 받았을 경우 present변수에 present라는 이름의 쿠키값을 가져와저장하고 만약 present값이 'money'이고 접속한 host의 주소가 `127.0.0.1`일 경우 flag라는 쿠키에 FLAG값을 넣어 설정해준다.

```
memo_text = ""

@app.route("/memo")
def memo():
    global memo_text
    text = request.cookies.get('flag', '')
    memo_text += text + "\n"
    return render_template("memo.html", memo=memo_text)
```

해당 코드는 `/memo`url에 접속했을 경우 실행이된다. memo_text를 글로벌변수로 선언하고 text변수에 flag라는 이름의 쿠키값을 저장한다. memo_text 글로벌 변수에 저장한 text값과 한줄을 띄는 문자를 저장하고 html에서 출력을 시켜준다.

접근방식

우선 FLAG를 얻는 코드부터 역순으로 찾아봐야 한다. `/memo`url을 보면 flag라는 쿠키값 을 가져오는데 flag의 쿠키값이 설정이 되있어야 flag가 출력이 된다. 그렇다면 flag쿠키값을 설정하는 부분을 찾아야한다. flag쿠키값을 설정하는 코드는 `/check-present`부분에 있다. 해당 코드에서 present의 쿠키값이 money이고 `127.0.0.1`의 host로 접속을 했을 경우 설정이 된다. 따라서 read_url로 서버에게 해당 url에 접속을 시켜야 flag를 얻을 수 있다는 것을 알 수 있고, present의 쿠키값이 설정된 부분을 찾아야한다. 하지만 present의 쿠키값을 설정하는 부분이 존재하지 않는데, `/present`url 코드를 보면 불필요한 코드가 있는 것을 확인할 수 있는데 바로 헤더를 설정하는 것이다. 서버에서 쿠키값을 설정할 때 헤더에 `Set-Cookie: {key}={value}` 형식으로 response를 내면 설정이 된다. 따라서 {sender}:{present} 의 형식으로 헤더가 설정이 되고있으니 이 입력값을 이용해 present 쿠키값을 money로 설정해주면 된다.

코드작성

sender = "Set-Cookie" present = "present = money"

문제 해결

flag를 얻기위해서 `/give-presnet`url에 접속해 sender에 "Set-Cookie"를 입력하고 present에 "present = money"를 입력해 POST요청을 날리면 `/memo`에서는 어떤 컴퓨터에서 요청한것 상관없이 모든 flag쿠키값을 출력시키기때문에 해당 url에 접속하면 flag가 출력이 된다.