

# BÁO CÁO THỰC HÀNH

Môn học: Bảo mật web và ứng dụng

Lab 2: Tổng quan các lỗ hổng bảo mật web thường gặp (phần 2)

GVHD: Nghi Hoàng Khoa

## 1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT213.P11.ATCL.1

STT	Họ và tên	MSSV	Email
1	Tôn Thất Bình	21520639	2152xxxx@gm.uit.edu.vn
2	Nguyễn Văn Hào	20521293	2052xxxx@gm.uit.edu.vn

## 2. NỘI DUNG THỰC HIỆN:<sup>1</sup>

STT	Công việc	Kết quả tự đánh giá
1	Bài tập 1	100%
2	Bài tập 2	100%
3	Bài tập 3	100%
4	Bài tập 4	100%
5	Bài tập 5	100%

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

<sup>1</sup> Ghi nội dung công việc, các kịch bản trong bài Thực hành

## BÁO CÁO CHI TIẾT

### a) A06:2021 – Vulnerable and Outdated Components

**Bài tập 1:** Báo cáo lỗ hổng đang được thực hành.

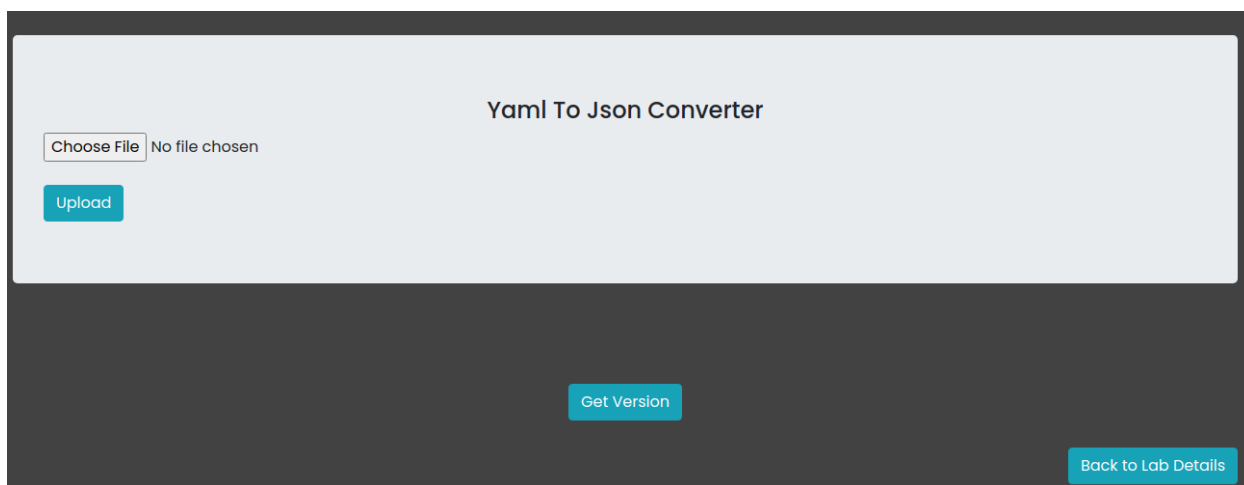
**#Tiêu đề:** Sử dụng các thành phần lỗi thời

**# Mô tả lỗ hổng:**

**### Tóm tắt:** Thực hiện chuyển đổi file .yaml sang file .json và thực hiện truyền file yaml có code thực thi

**### Các bước thực hiện lại và bằng chứng:**

- **Bước 1:** Truy cập vào trang web ta biết được trang web có chức năng chuyển đổi file định dạng yaml thành json.



- **Bước 2:** Để khai thác lỗ hổng ta tạo file yaml chứa đoạn code khai thác nhằm liệt kê các thành phần có trong thư mục hiện tại



- Từ kết quả web trả về có thể thấy đoạn lệnh ls đã được thực thi và in ra các thành phần có trong thư mục hiện tại:

```
<h5>
  Here is your output:
</h5>
<br>
<pre>
  p&#x27;Dockerfile\nProcfile\nSolutions\napp.log\ndb.sqlite3\ndb.sqlite3~f1cf11156c
  656314790387c2c9eb7f187a3d480e\ndocker-compose.yml\nintroduction\nmanage.py\npygoa
  t\nrequirements.txt\nruntime.txt\nstaticfiles\ntest.log\n&#x27;
</pre>
<br>
<b>
  Check Django Terminal for Command's output
</b>
```

# **Mức độ ảnh hưởng:** Rất cao

# **Khuyến cáo:**

- Sử dụng phiên bản mới nhất của thư viện xử lý YAML
- Thực hiện kiểm tra và lọc đầu vào nghiêm ngặt
- Giới hạn các tính năng YAML được phép sử dụng
- Triển khai sandbox để cô lập quá trình xử lý YAML
- Thường xuyên cập nhật và kiểm tra bảo mật cho tất cả các thành phần của hệ thống

## b) A07:2021 – Identification and Authentication Failures

**Bài tập 2:** Báo cáo lỗ hổng đang được thực hành.

#**Tiêu đề:** Lỗi nhận dạng và xác thực.

# **Mô tả lỗ hổng:**

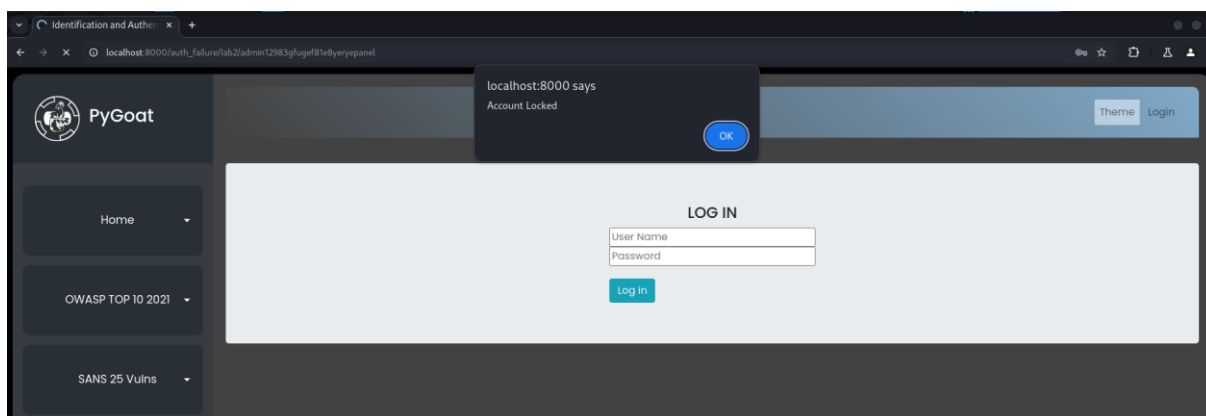
### **Tóm tắt:** Có user và password ở dạng hash. Ta chỉ cần đăng nhập nhiều lần (5 lần) thực hiện phá hoại để chặn tài khoản truy cập 1 ngày.

### **Các bước thực hiện lại và bằng chứng:**

- **Bước 1:** Ta đăng nhập vào tài khoản xác thực quản trị đã được cấp với user: admin và password chưa biết trước:



- **Bước 2:** Ta có thể thử các password thông dụng như là: admin, admin123, admin123@, adm1n@, adm1n123 nhưng sau khi thử 5 lần sai liên tiếp thì tài khoản của ta đã bị khoá 1440 phút. Như vậy ta đã thực hiện khoá thành công tài khoản admin:



```
elif request.method == "POST":
    username = request.POST["username"]
    password = request.POST["password"]
    try:
        user = AF_admin.objects.get(username=username)
        print(type(user.lockout_cooldown))
        if user.is_locked == True and user.lockout_cooldown > datetime.date.today():
            return render(request, "Lab_2021/A7_auth_failure/lab2.html", {"is_locked":True})

    try:
        ph = PasswordHasher()
        ph.verify(user.password, password)
        if user.is_locked == True and user.lockout_cooldown < datetime.date.today():
            user.is_locked = False
            user.last_login = datetime.datetime.now()
            user.failattempt = 0
            user.save()
            return render(request, "Lab_2021/A7_auth_failure/lab2.html", {"user":user, "success":True, "failure":False})
    except:
        fail_attempt = user.failattempt + 1
        if fail_attempt == 5:
            user.is_active = False
            user.failattempt = 0
            user.is_locked = True
            user.lockout_cooldown = datetime.datetime.now() + datetime.timedelta(minutes=1440)
            user.save()
            return render(request, "Lab_2021/A7_auth_failure/lab2.html", {"user":user, "success":False, "failure":True, "is_locked":True})
        user.failattempt = fail_attempt
        user.save()
        return render(request, "Lab_2021/A7_auth_failure/lab2.html", {"success":False, "failure":True})
except Exception as e:
    print(e)
    return render(request, "Lab_2021/A7_auth_failure/lab2.html", {"success":False, "failure":True})
```

[View Code](#)[Back to Lab Details](#)

# **Mức độ ảnh hưởng:** trung bình đến cao

# **Khuyến cáo:**

- Triển khai các biện pháp chống tấn công brute-force tinh vi hơn, như CAPTCHA sau lần thử sai thứ 2 hoặc 3.
- Sử dụng xác thực đa yếu tố (MFA) để tăng cường bảo mật.
- Theo dõi và phân tích mẫu đăng nhập để phát hiện hoạt động bất thường.
- Cân nhắc sử dụng thời gian chờ tăng dần thay vì khóa tài khoản hoàn toàn.
- Thông báo cho người dùng khi có nỗ lực đăng nhập bất thường vào tài khoản của họ.

### c) A08:2021 – Software and Data Integrity Failures

**Bài tập 3:** Báo cáo lỗ hổng đang được thực hành.

#**Tiêu đề:** Lỗi về tính toàn vẹn của phần mềm và dữ liệu

# **Mô tả lỗ hổng:**

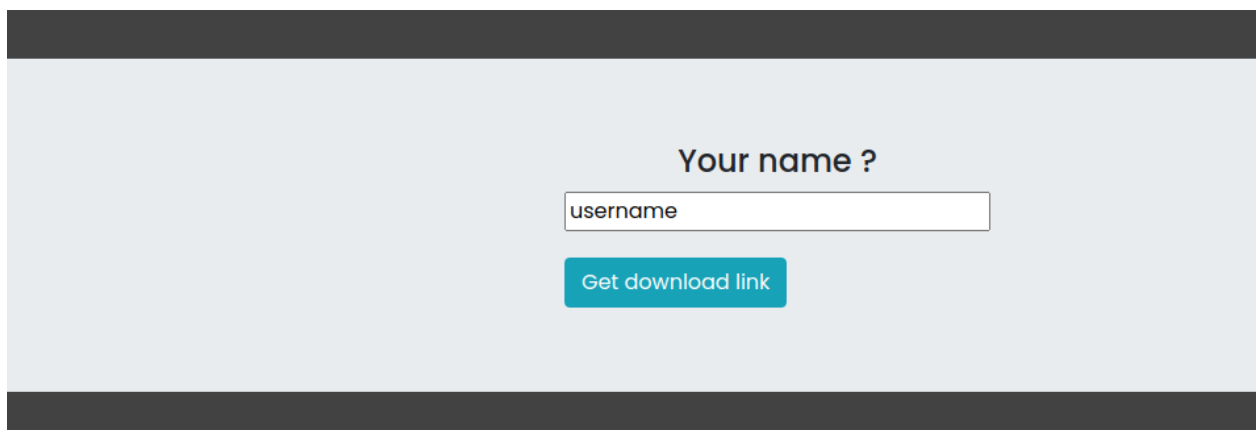
### **Tóm tắt:** Tấn công xss để lừa người dùng tải về file giả mạo.

### **Các bước thực hiện lại và bằng chứng:**

- **Bước 1:** Sau khi truy cập vào đường dẫn đầu tiên ta có thể thấy đoạn script dùng để tải về file fake.txt

```
<h1>
  Hey user <script>
    document.getElementById("download_link").setAttribute("href","/static/fake.txt");
  </script>
  user <script>
    document.getElementById("download_link").setAttribute("href","/static/fake.txt");
  </script>
  ,
</h1>
```

- **Bước 2:** Đường dẫn thứ 2 dẫn ta đến trang yêu cầu nhập tên người dùng và sau đó thực hiện lấy đường dẫn tải file



- Ta nhận thấy tên người dùng ta nhập vào sẽ được hiển thị trong thẻ h1 cùng với đường dẫn tải về file real.txt

```
Here is your download <a id="download_link" href="/static/real.txt" style="color:#ff5"
download>
  Link
</a>
<h1>
  Hey username,
</h1>
```

- **Bước 3:** Thay vì nhập tên người dùng ta sẽ nhập vào đoạn script js dưới đây để thay đổi đường dẫn tải file từ real.txt thành fake.txt. hàm window.onload giúp đoạn code được thực thi ngay khi vào trang

```
<script>
  function changeDownloadLink() {
    const downloadLink = document.getElementById('download_link');
    downloadLink.href = '/static/fake.txt';
    downloadLink.download = 'fake.txt';
  }
  window.onload = changeDownloadLink;
</script>
```

- Kết quả cho thấy đoạn script ta nhập vào đã được chèn vào trong thẻ h1

```
Here is your download <a id="download_link" href="/static/real.txt" style="color:#ff5"
download>
  Link
</a>
<h1>
  Hey <script>
    function changeDownloadLink() {
      const downloadLink = document.getElementById('download_link');
      downloadLink.href = '/static/fake.txt';
      downloadLink.download = 'fake.txt';
    }
    window.onload = changeDownloadLink;
  </script>
</h1>
```

- Kết quả sau khi nhấp vào đường link ta sẽ tải về file fake.txt thay vì real.txt

**Response**

PrettyRawHexRender

1 HTTP/1.1 200 OK

2 Server: gunicorn

3 Date: Tue, 01 Oct 2024 02:08:29 GMT

4 Connection: close

5 Content-Type: text/plain; charset="utf-8"

6 Cache-Control: max-age=0, public

7 Access-Control-Allow-Origin: \*

8 Last-Modified: Fri, 02 Sep 2022 06:45:48 GMT

9 ETag: "6311a69c-16"

10 Content-Length: 22

11

12 this is malicious file

### Mức độ ảnh hưởng: Cao

### # Khuyến cáo:

- Triển khai các kiểm tra tính toàn vẹn như chữ ký số trên bất kỳ đối tượng đã được tuần tự hóa nào để ngăn chặn việc tạo ra đối tượng độc hại hoặc can thiệp vào dữ liệu.
- Cách ly và chạy mã giải tuần tự hóa trong môi trường có quyền hạn thấp khi có thể.
- Ghi lại các ngoại lệ và thất bại trong giải tuần tự hóa, chẳng hạn như khi kiểu dữ liệu đến không phải là kiểu dữ liệu mong đợi, hoặc khi giải tuần tự hóa gây ra ngoại lệ.
- Hạn chế hoặc giám sát khả năng kết nối mạng đến và đi từ các container hoặc máy chủ thực hiện giải tuần tự hóa. Giám sát giải tuần tự hóa, cảnh báo nếu một người dùng thực hiện giải tuần tự hóa liên tục.

## d) A09:2021 – Security Logging and Monitoring Failures

**Bài tập 4:** Báo cáo lỗ hổng đang được thực hành.

**#Tiêu đề:** Ghi nhật ký bảo mật và giám sát lỗi

### # Mô tả lỗ hổng:

**### Tóm tắt:** Trang web ghi lại các log không cần thiết và khi bị lộ có thể dẫn đến tiết lộ thông tin nhạy cảm của người dùng.

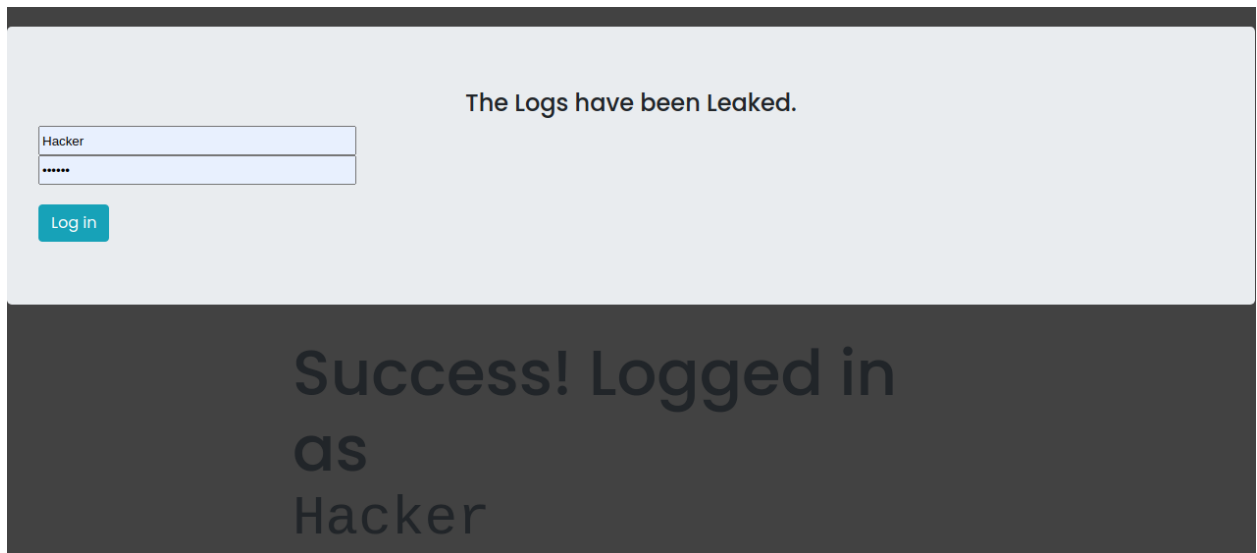
### ### Các bước thực hiện lại và bằng chứng:

- Bước 1: Được biết log của trang web đã bị lộ, ta truy cập theo route <http://127.0.0.1:8000/debug>, phát hiện thấy log của trang web và tài khoản cũng như password của một người dùng

```
INFO "GET /login/ HTTP/1.1" 200 7978
INFO "GET /a10_lab?username=Hacker&password=Hacker HTTP/1.1" 301 0
INFO "GET /logout HTTP/1.1" 301 0
```

- Bước 2: Sử dụng tài khoản và password đã tìm được, đăng nhập vào trang web và ta thu được kết quả bên dưới:





# **Mức độ ảnh hưởng:** Cao

# **Khuyến cáo:**

- Đảm bảo rằng các bản ghi (log) được tạo ra theo định dạng có thể dễ dàng sử dụng bởi các công cụ quản lý log trung tâm.
- Cần thiết lập hệ thống giám sát và cảnh báo hiệu quả để phát hiện và phản ứng kịp thời với các hoạt động đáng ngờ.
- Đảm bảo rằng không có thông tin nhạy cảm nào như mật khẩu được ghi lại.

### e) A10:2021 – Server-Side Request Forgery (SSRF)

**Bài tập 5:** Báo cáo lỗ hổng đang được thực hành.

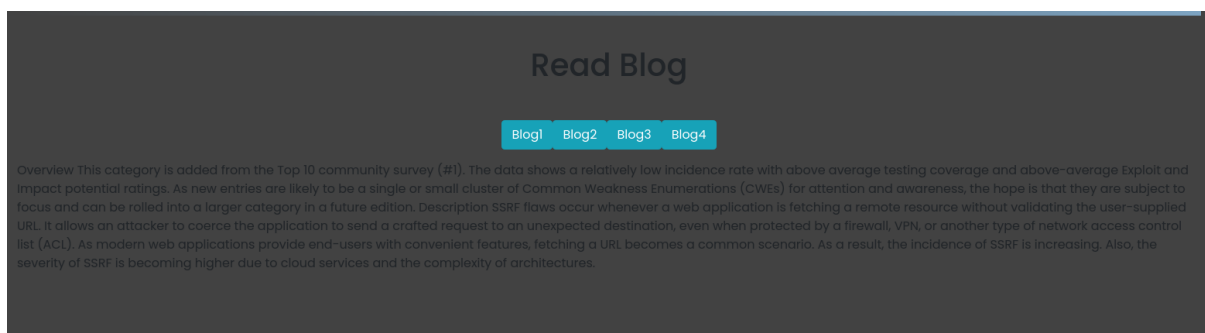
#**Tiêu đề:** SSRF

# **Mô tả lỗ hổng:**

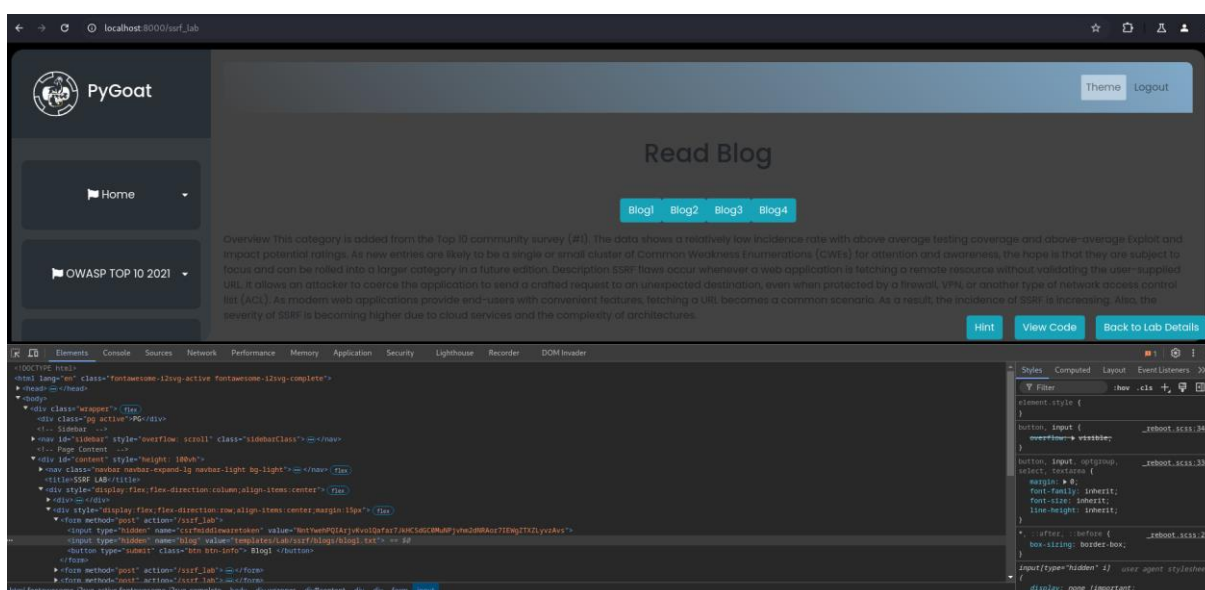
### **Tóm tắt:** Thực hiện thay đổi đường dẫn truy cập, kiểm tra button và dữ liệu sau khi thay đổi.

### **Các bước thực hiện lại và bằng chứng:**

- **Bước 1:** Ta xem nội dung của các Blog xong ta thử lấy Blog1 làm ví dụ:



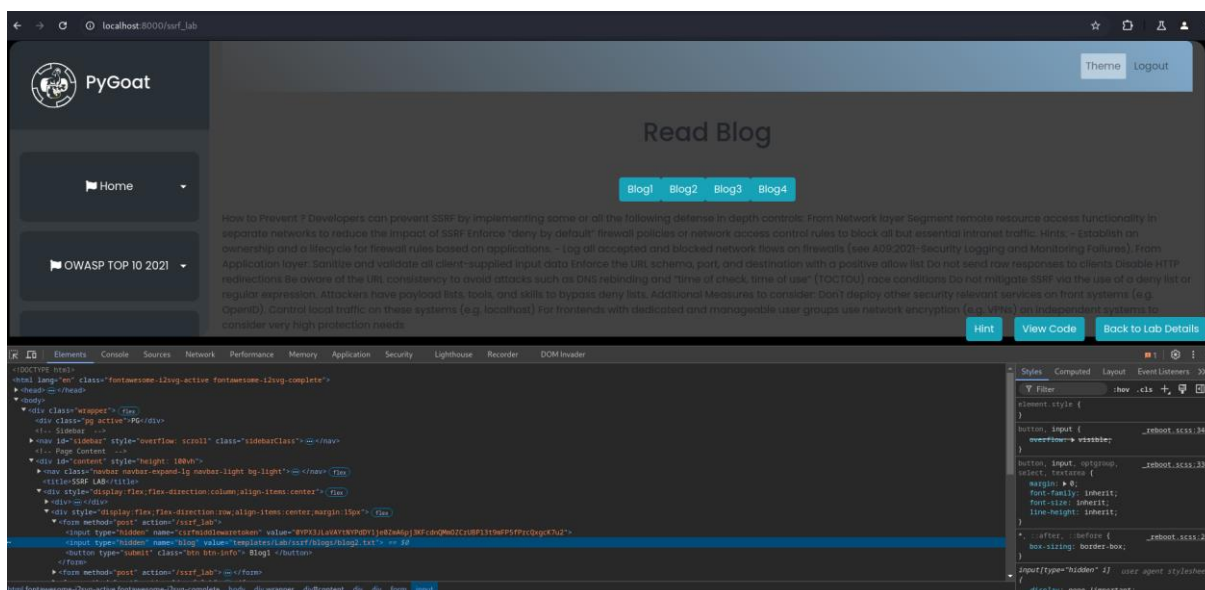
- **Bước 2:** Tiếp tục mở bảng Elements và thấy đường dẫn file Blog1:



- **Bước 3:** Sau đó ta thử đổi đường dẫn **/blog1.txt** thành **/blog2.txt** để kiểm tra thấy nội dung của Blog1 đã thay đổi thành nội dung Blog2.

## Lab 2: Tổng quan các lỗ hổng bảo mật web thường gặp (phần 2)

11



- Bài yêu cầu thay đổi đường dẫn truy cập nên là ta thử file python: **urls.py**

- **Bước 4:** Thay đường dẫn **templates/Lab/ssrf/blogs/blog1.txt** thành **urls.py**, ta thu được kết quả bên dưới:

```
<input type="hidden" name="blog" value="urls.py"> == $0
<button type="submit" class="btn btn-info"> Blog1 </button>
```



- Ta có gợi ý của bài là:

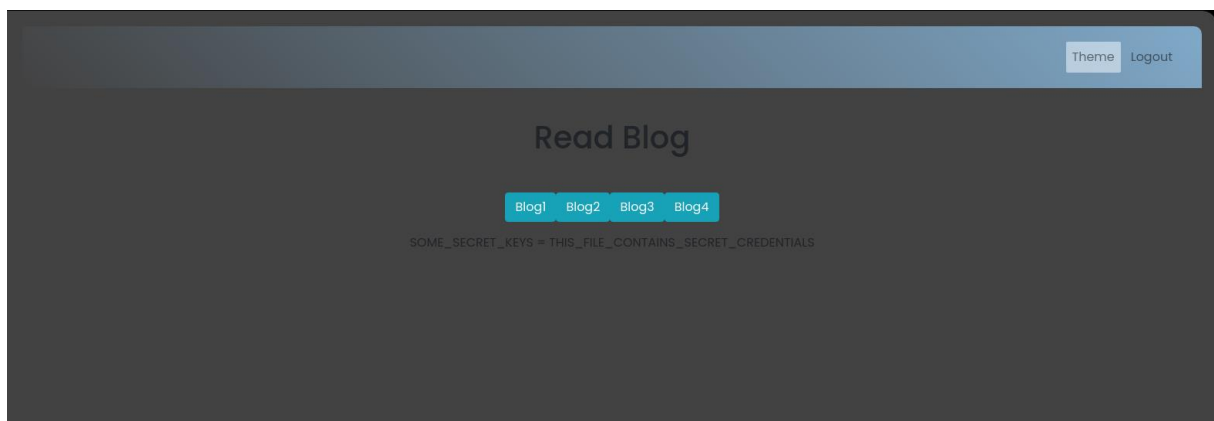


- Để clean code thì thông thường file .env sẽ ở phía trước các folder chứa code python vậy nên phần truyền vào sẽ là `../.env`

- **Bước 5:** Thay đổi đường dẫn `urls.py` thành `../.env`

```
<input type="hidden" name="blog" value="../.env"> == $0
<button type="submit" class="btn btn-info"> Blog1 </button>
```

- Kết quả thu được:



# **Mức độ ảnh hưởng:** Cao

# **Khuyến cáo:**

- Triển khai danh sách cho phép (allowlist) cho các domain và IP được phép truy cập.
- Sử dụng các thư viện xử lý URL an toàn để ngăn chặn việc thao túng URL.
- Thực hiện kiểm tra và xác thực nghiêm ngặt đối với tất cả các đầu vào người dùng, đặc biệt là các URL và đường dẫn.
- Sử dụng tường lửa ứng dụng web (WAF) để phát hiện và chặn các yêu cầu đáng ngờ.
- Giới hạn kết nối ra ngoài từ máy chủ ứng dụng.
- Sử dụng các kỹ thuật như DNS pinning để ngăn chặn việc phân giải DNS động.

## YÊU CẦU CHUNG

1. Sinh viên tìm hiểu và thực hiện bài tập theo yêu cầu, hướng dẫn.
2. Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
3. Sinh viên báo cáo kết quả thực hiện và nộp bài.

### Báo cáo:

- File **.DOCX và .PDF**. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng **Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.**
- Đặt tên theo định dạng: [Mã lớp]-ExeX\_GroupY. (trong đó X là Thứ tự Bài tập, Y là mã số thứ tự nhóm trong danh sách mà GV phụ trách công bố).  
*Ví dụ: [NT101.K11.ANTT]-Exe01\_Group03.*
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- **Không đặt tên đúng định dạng – yêu cầu, sẽ KHÔNG chấm điểm bài nộp.**
- Nộp file báo cáo trên theo thời gian đã thống nhất tại [courses.uit.edu.vn](https://courses.uit.edu.vn).

### Đánh giá:

- Hoàn thành tốt yêu cầu được giao.
- Có nội dung mở rộng, ứng dụng.

*Bài sao chép, trộm, ... sẽ được xử lý tùy mức độ vi phạm.*

**HẾT**