Phòng thí nghiệm An toàn Thông tin
Trường Đại học Công nghệ Thông tin (UIT)

BÁO CÁO THỰC HÀNH

Môn học: Bảo mật Web và Ứng dụng
Kỳ báo cáo: Buổi 02 (Session 02)
Tên chủ đề: Tổng quan các lỗ hổng bảo mật web thường gặp (phần 2)
GVHD: Ngô Đức Hoàng Sơn
Ngày báo cáo: 4/3/2024
Nhóm: 09

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)
Lớp: NT213.O22.ATCL.2

STT	Họ và tên	MSSV	Email
1	Hồ Ngọc Thiện	21522620	21522620@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Công việc	Kết quả tự đánh giá
1	Kịch bản 01	100%
2	Kịch bản 02	100%
3	Kịch bản 03	100%
4	Kịch bản 04	100%
5	Kịch bản 05	100%

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

[1] Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

1. Kịch bản 01

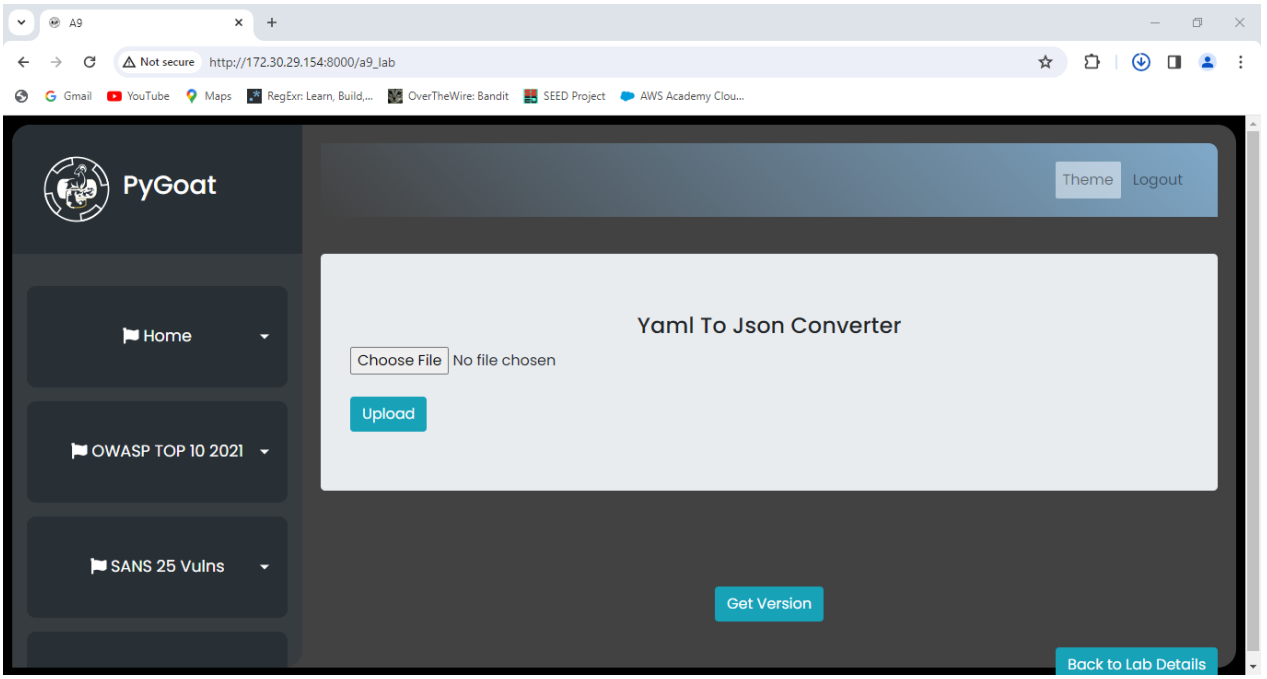
Tiêu đề: Vulnerable and Outdated Components. Tài sản bị ảnh hưởng: dữ liệu, thông tin

Mô tả lỗ hổng:

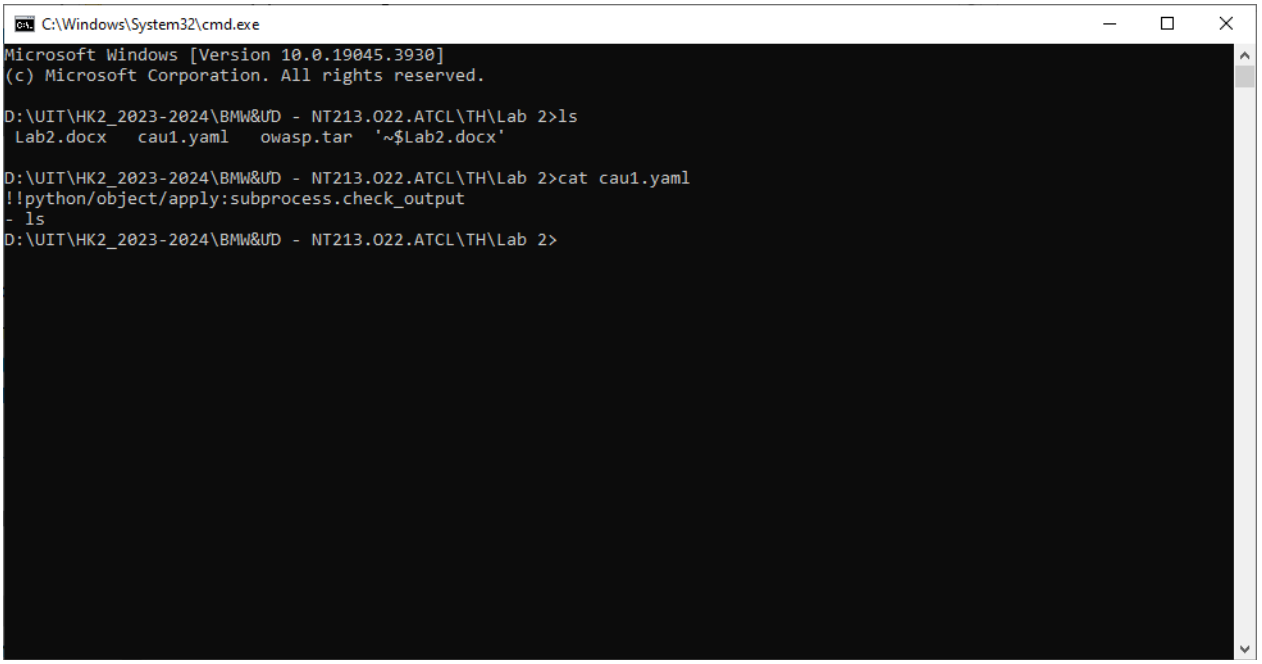
Lỗ hổng "Vulnerable and Outdated Components" trong ứng dụng web là một vấn đề phổ biến khi ứng dụng sử dụng các thành phần (components) cũ và có lỗ hổng bảo mật, hoặc không được cập nhật đầy đủ. Trong trường hợp này, việc sử dụng một công cụ chuyển đổi từ định dạng YAML sang JSON mà không kiểm tra dữ liệu đầu vào cẩn thận, cho phép tấn công phụ thuộc vào việc thực thi mã từ tệp YAML.

Các bước thực hiện:

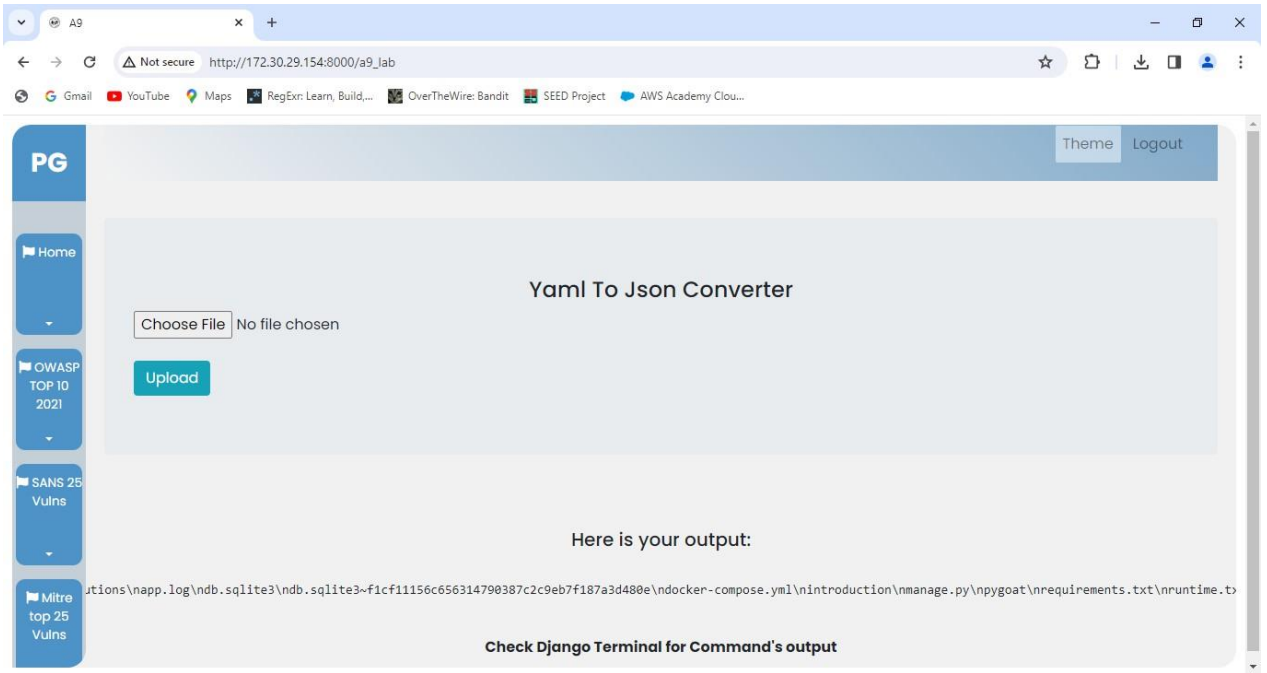
Đầu tiên truy cập bài thực hành tại: http://localhost:8000/a9. Đây là một trang dùng để chuyển dữ liệu với định dạng YAML sang JSON.



Ta sẽ thử khai thác bằng cách gửi một file yaml lên để thực thi code. Tạo file cau1.yaml có nội dung như bên dưới



Tải file cau1.yaml lên trang web ta thu được kết quả:



Mức độ ảnh hưởng của lỗ hổng: high

Tác động bảo mật nào mà kẻ tấn công có thể đạt được: Kẻ tấn công có thể thực thi code độc hại để tìm kiếm thông tin bảo mật được lưu trữ trên server.

Khuyến cáo khắc phục:

- Cập nhật components: Đảm bảo rằng tất cả các omponent của ứng dụng, bao gồm cả thư viện và framework, đều được cập nhật lên phiên bản mới nhất.

- Kiểm tra mã đầu vào: Thực hiện kiểm tra nghiêm ngặt dữ liệu đầu vào từ người dùng, đặc biệt là các dữ liệu có thể được chuyển đổi hoặc xử lý bởi các thành phần cũ hay có lỗ hổng.

- Giới hạn quyền truy cập

- Sử dụng thư viện chuyển đổi an toàn

Tài liệu tham khảo:

https://qawerk.com/blog/vulnerable-and-outdated-components/

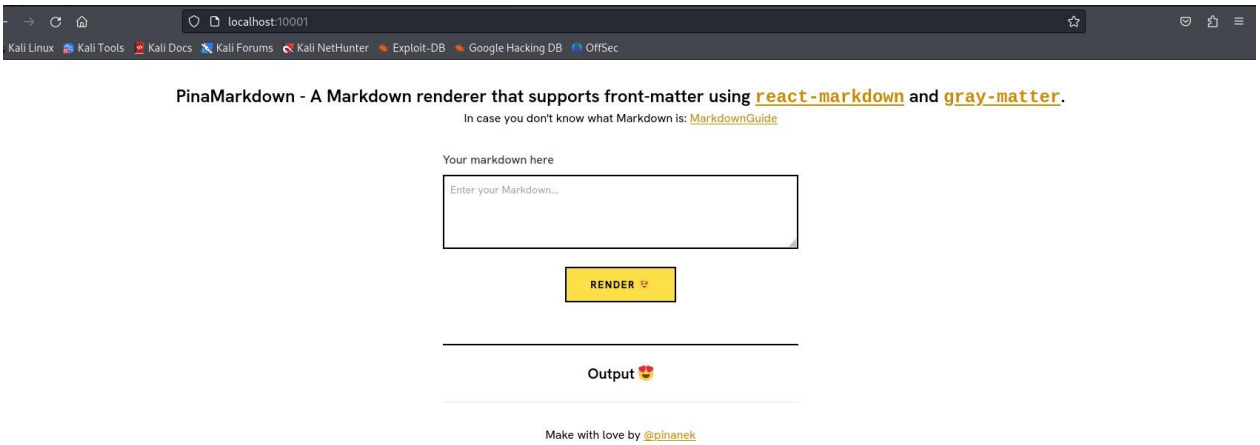
Bài tập 1: Thực hiện việc khai thác lỗ hổng với một ứng dụng render Markdown thành HTML. Sử dụng format sau mẫu để trình bày

Mô tả lỗ hổng:

Trang web cho phép render Markdown thành HTML. Tuy nhiên trang web sử dụng grey-matter phiên bản cũ và có lỗi liên quan tới thực thi code khi input.

Các bước để thực hiện:

Khi vào trang web ta thấy trang web giới thiệu được viết bằng react-markdown và gray-matter.



Khi tìm kiếm thông tin lỗ hổng về 2 thư viện này, ta tìm thấy CVE liên quan tới gray-matter

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed, or concur with the facts presented on these sites. Further, NIST does not endorse any commercial products that may be mentioned on these sites. Please address comments about this page to nvd@nist.gov.

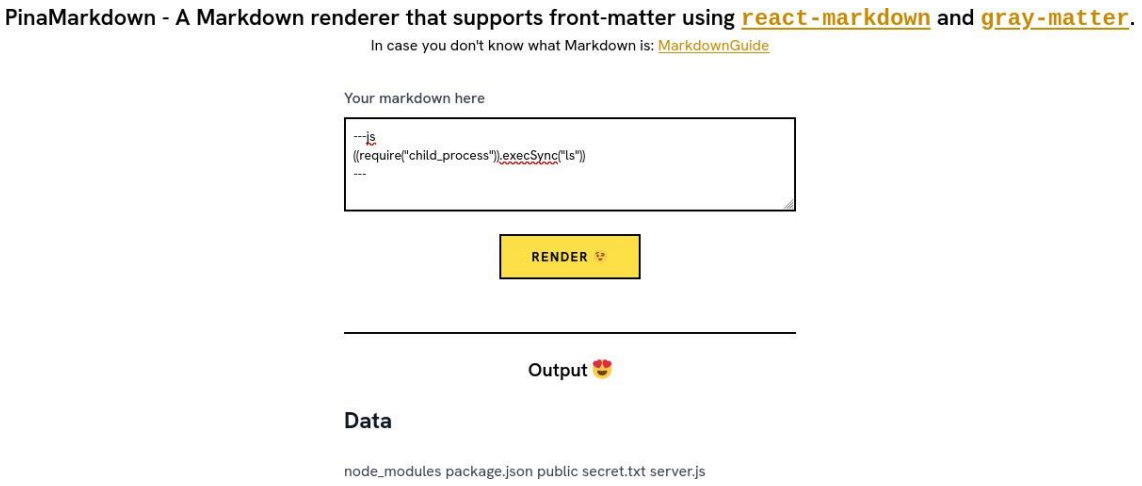
Hyperlink	Resource
https://github.com/gatsbyjs/gatsby/security/advisories/GHSA-7ch4-rr99-cqcw	Exploit Third Party Advisory

Weakness Enumeration

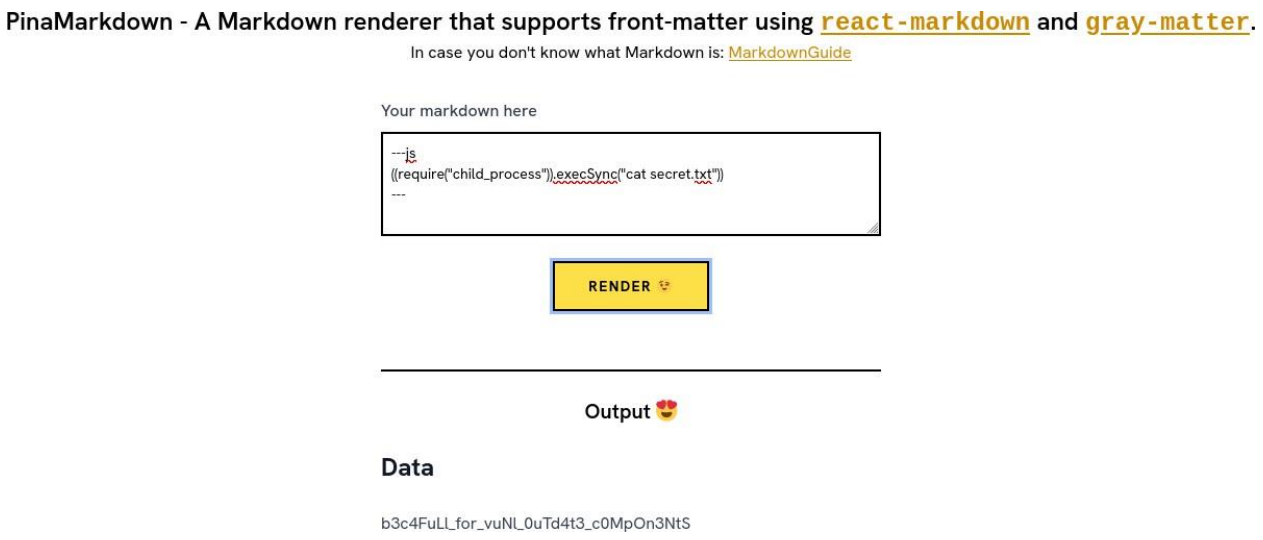
CWE-ID	CWE Name	Source
CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	NIST
CWE-20	Improper Input Validation	GitHub, Inc.
CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	GitHub, Inc.

Sử dụng POC đã tìm được và đổi lệnh thực thi thành “ls”.

```
---js
((require("child_process")).execSync("id >> /tmp/rce"))
---
```



Ta có thể thấy kết quả hiển thị 1 file là secret.txt, tiến hành đọc thử file này.



Mức độ ảnh hưởng của lỗ hổng: high.

Tác động bảo mật nào mà kẻ tấn công có thể đạt được: Kẻ tấn công có thể thực thi code độc hại để tìm kiếm thông tin bảo mật được lưu trữ trên server.

Khuyến cáo khắc phục:

Cập nhật phần mềm thường xuyên. Có filter khi upload file lên server, chặn quyền thực thi của file

2. Kịch bản 02

Tiêu đề: Identification and Authentication Failures. Tài sản bị ảnh hưởng: quyền truy cập tài khoản

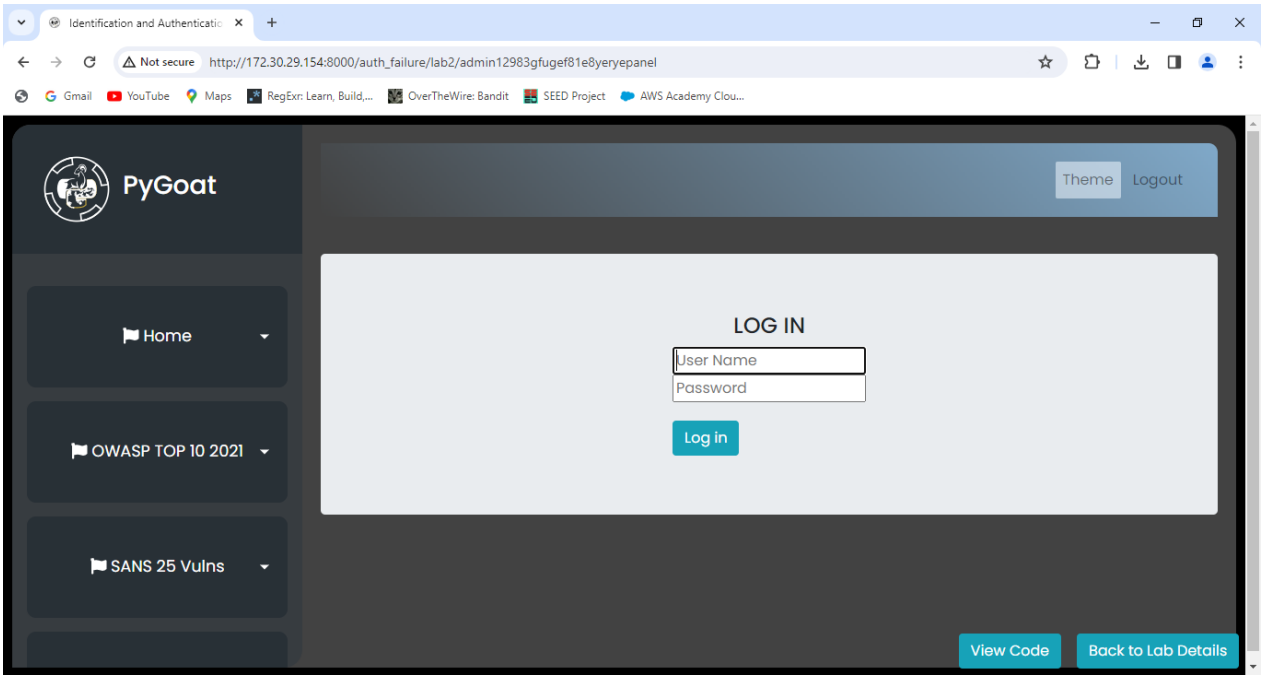
Mô tả lỗ hổng:

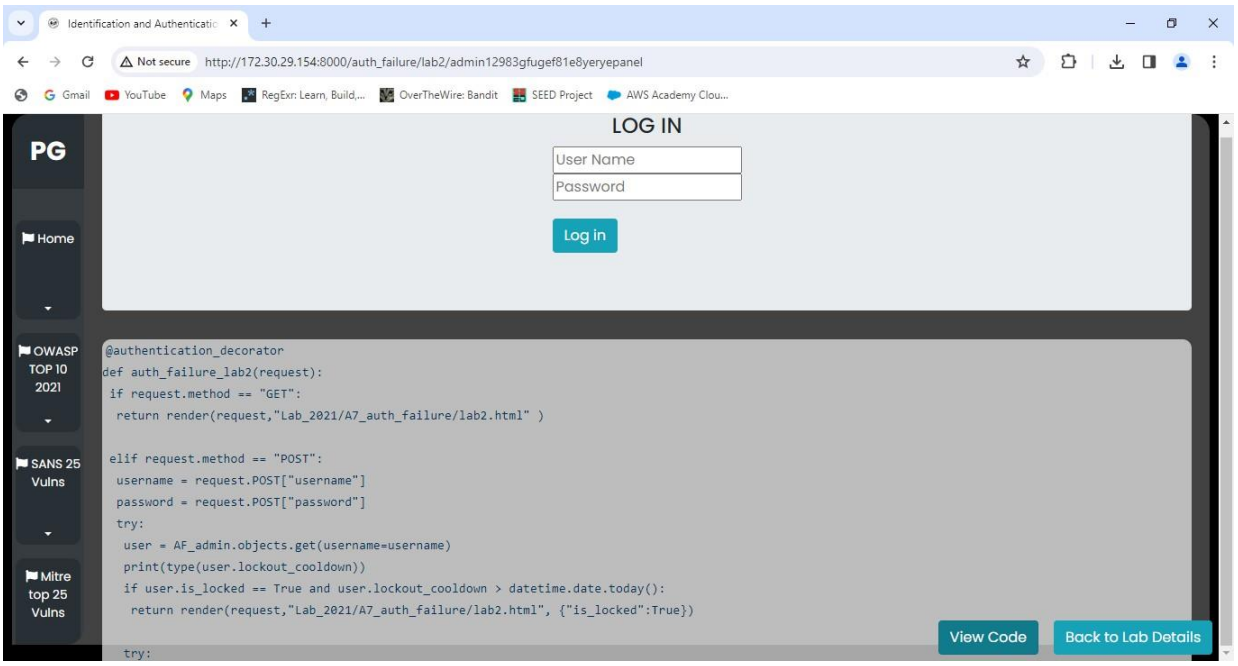
Lỗ hổng "Identification and Authentication Failures" trong ứng dụng web xảy ra khi hệ thống không thực hiện đủ biện pháp xác thực và nhận dạng người dùng một cách an toàn. Trong trường hợp này, trang web cung cấp tài khoản admin và mật khẩu được lưu dưới dạng hash. Tuy không đăng nhập nhưng attacker vẫn có thể thực hiện các cuộc tấn công phá hoại để chặn tài khoản admin truy cập trong một khoảng thời gian nhất định.

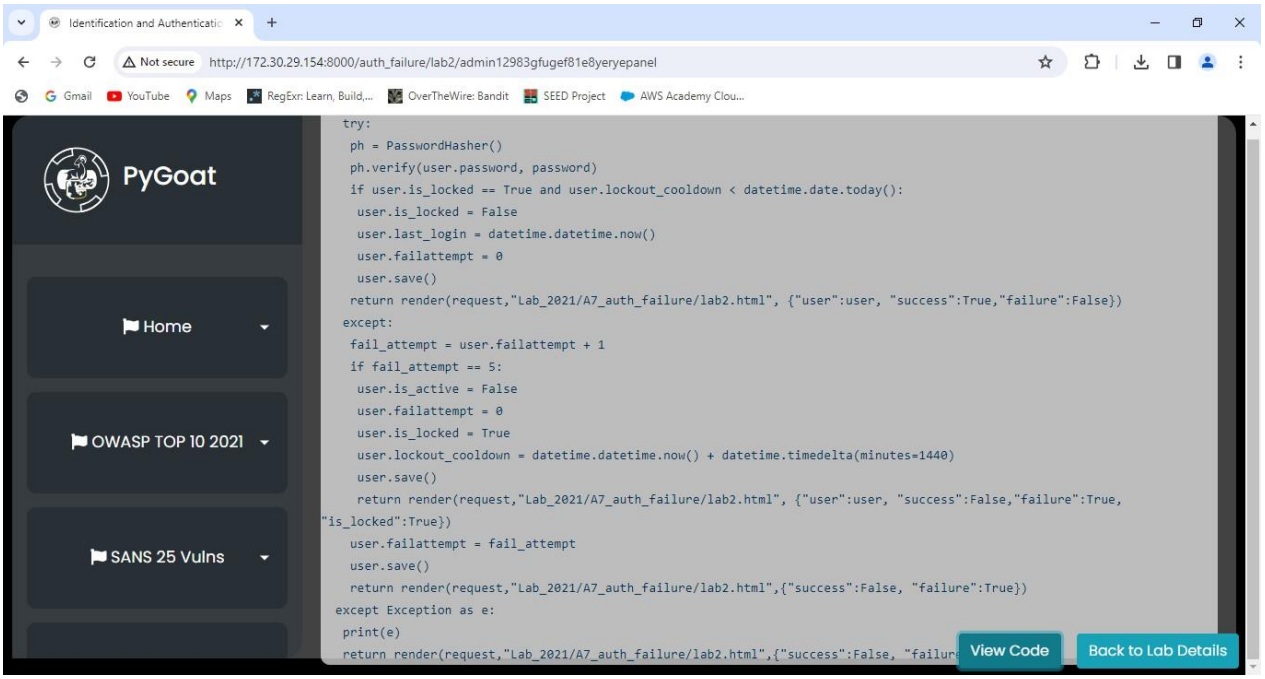
Các bước thực hiện:

Đầu tiên truy cập kịch bản tại:

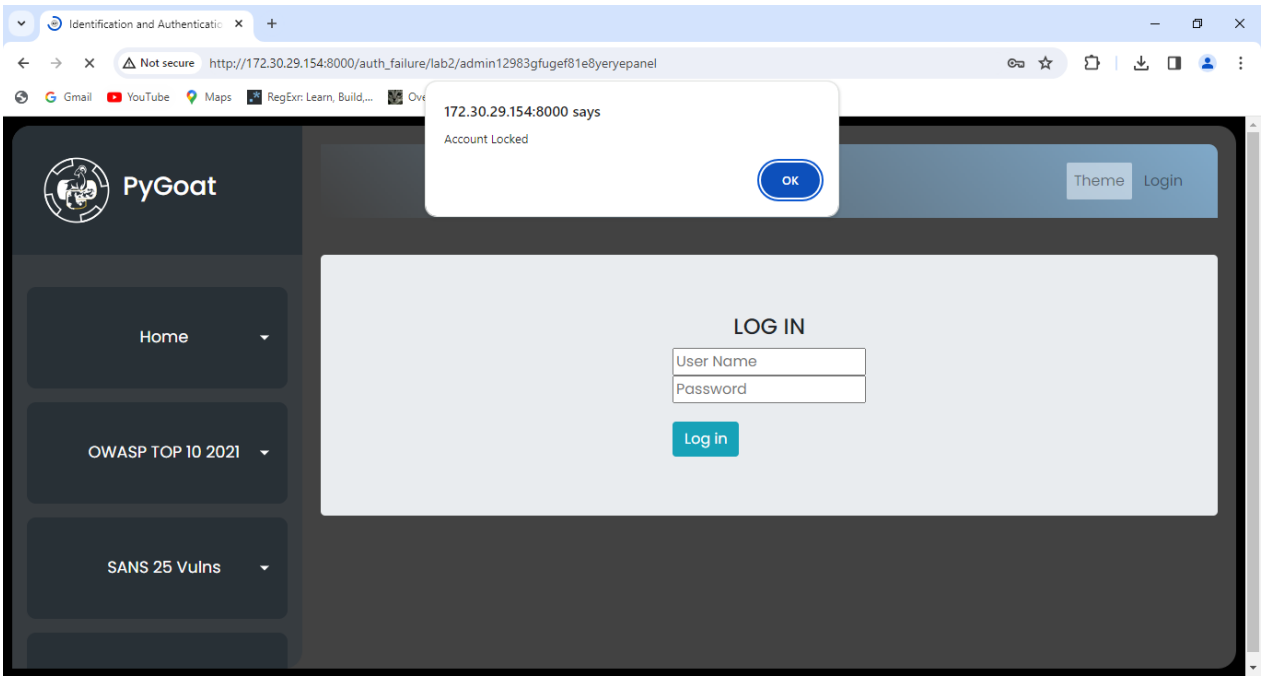
http://localhost:8000/auth_failure/lab2/admin12983gfugef81e8yeryepanel


Identification and Authenticatic
Not secure
http://172.30.29.154:8000/auth_failure/lab2/admin12983gfugef81e8yeryepanel
Gmail
YouTube
Maps
RegExr: Learn, Build,...
OverTheWire: Bandit
SEED Project
AWS Academy Clou...
PyGoat
Theme
Logout
Home
OWASP TOP 10 2021
SANS 25 Vulns
LOG IN
User Name
Password
Log in
View Code
Back to Lab Details


Identification and Authenticatic
Not secure
http://172.30.29.154:8000/auth_failure/lab2/admin12983gfugef81e8yeryepanel
Gmail
YouTube
Maps
RegExr: Learn, Build,...
OverTheWire: Bandit
SEED Project
AWS Academy Clou...
PG
Home
OWASP TOP 10 2021
SANS 25 Vulns
Mitre top 25 Vulns
LOG IN
User Name
Password
Log in
@authentication_decorator
def auth_failure_lab2(request):
if request.method == "GET":
return render(request,"Lab_2021/A7_auth_failure/lab2.html")
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:
View Code
Back to Lab Details



Từ code, ta thử password sai 5 lần thì tài khoản sẽ bị khóa 1440 phút = 24 giờ. Vậy ta thực hiện khóa thành công tài khoản admin



Mức độ ảnh hưởng: high

Tác động bảo mật nào mà kẻ tấn công có thể đạt được: Kẻ tấn công có thể tiến hành tấn công DOS vào trang web và chặn quyền truy cập của các user bình thường.

Khuyến cáo khắc phục:

- Sử dụng các phương pháp xác thực mạnh mẽ như hai yếu tố xác thực (2FA) hoặc mã thông báo xác thực một lần (OTP).

- Thực hiện quản lý tài khoản bằng cách theo dõi, kiểm soát và quản lý quyền truy cập của tài khoản admin. Đảm bảo rằng các tài khoản không sử dụng hoặc có nguy cơ bị chiếm đoạt được vô hiệu hóa hoặc xóa bỏ.

Tài liệu tham khảo:

https://cyolo.io/blog/identification-and-authentication-failures-and-how-to-prevent-them/

3. Kịch bản 03

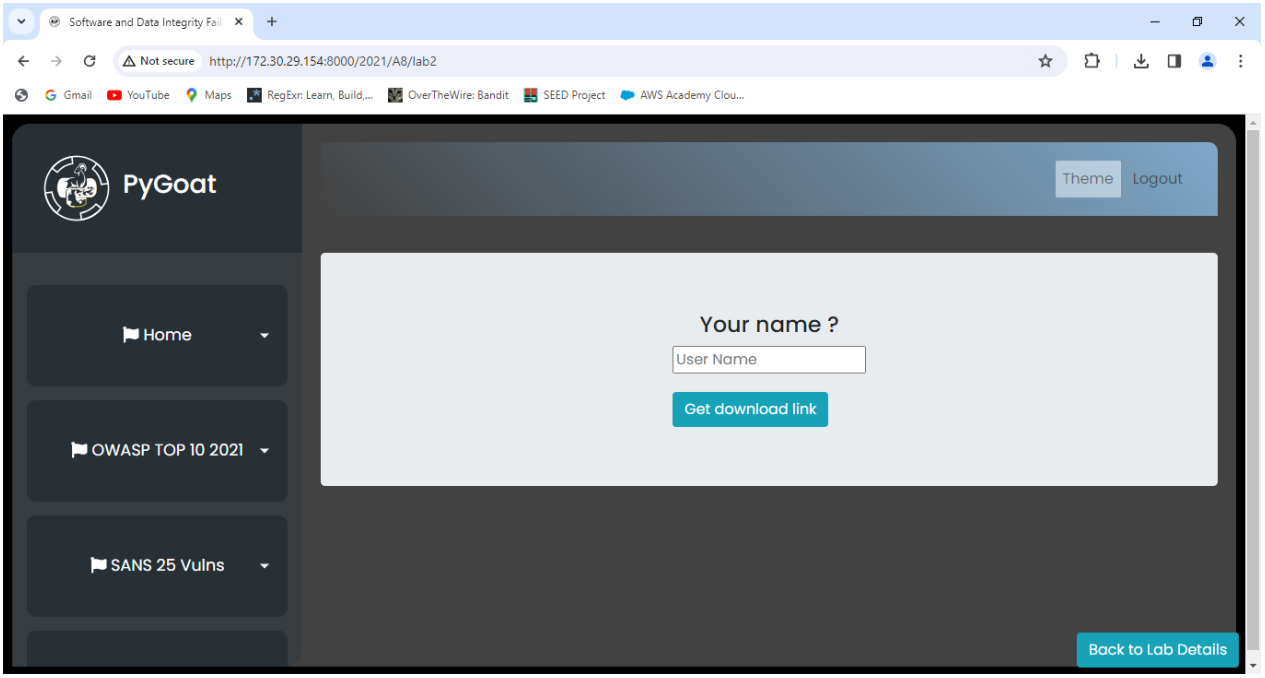
Tiêu đề: Software and Data Integrity Failures. Tài sản bị ảnh hưởng: dữ liệu, tính toàn vẹn

Mô tả lỗ hổng:

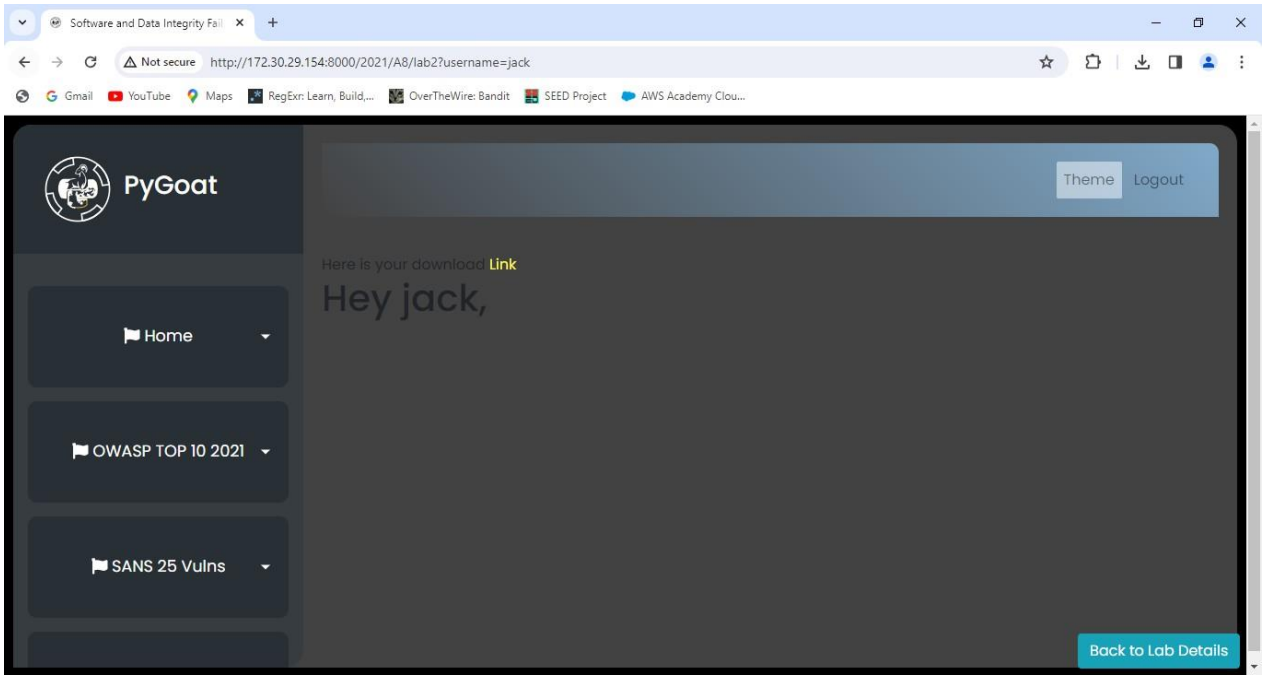
Lỗ hổng "Software and Data Integrity Failures" xảy ra khi người tấn công tận dụng các lỗ hổng trong phần mềm hoặc quy trình để can thiệp vào tính toàn vẹn của dữ liệu. Trong trường hợp này, khi người dùng truy cập vào trang web và tải xuống một tệp dữ liệu, kẻ tấn công sẽ thực hiện việc chỉnh sửa nội dung của tệp này, sau đó tải lên một tệp mới đã được chỉnh sửa để can thiệp vào tính toàn vẹn dữ liệu.

Các bước thực hiện:

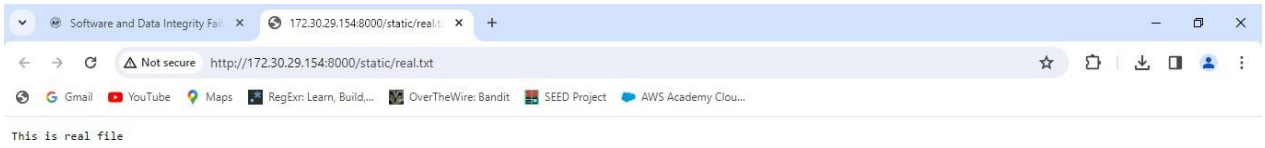
Đầu tiên ta truy cập vào http://localhost:8000/2021/A8/lab2, ta có thể thấy thông tin để nhập và tải file.



Nhập một tên người dùng bất kỳ thì ta thu được link download.

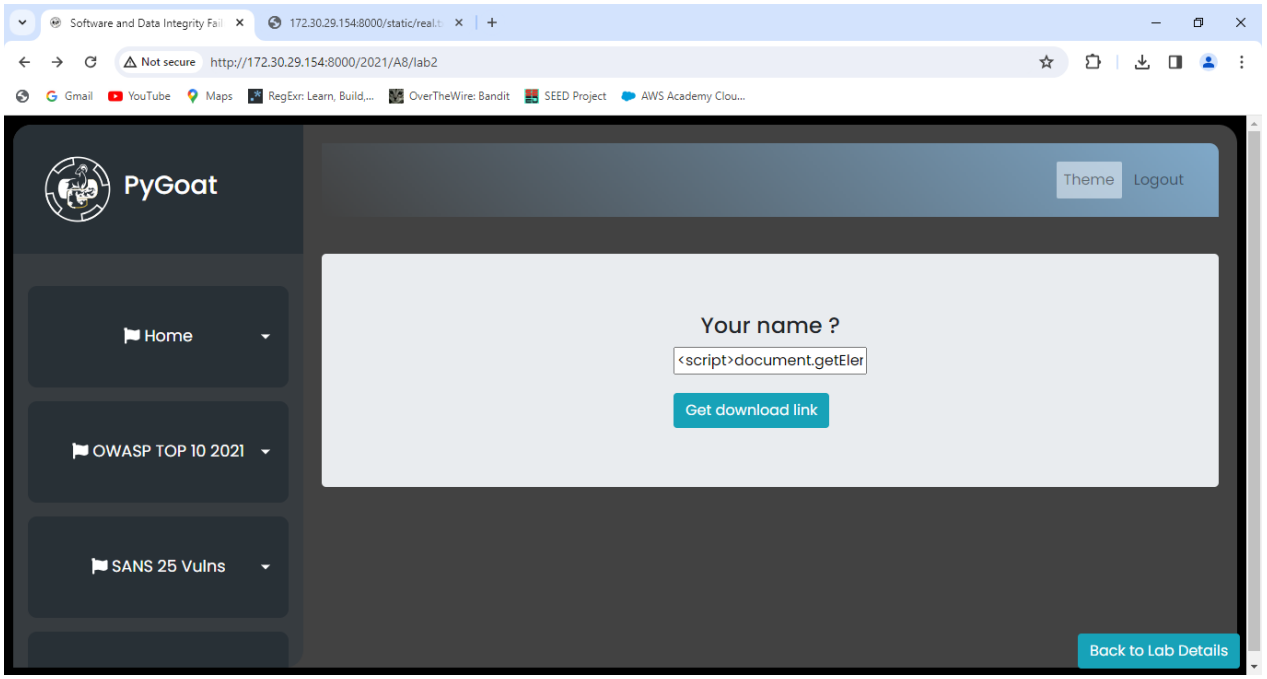


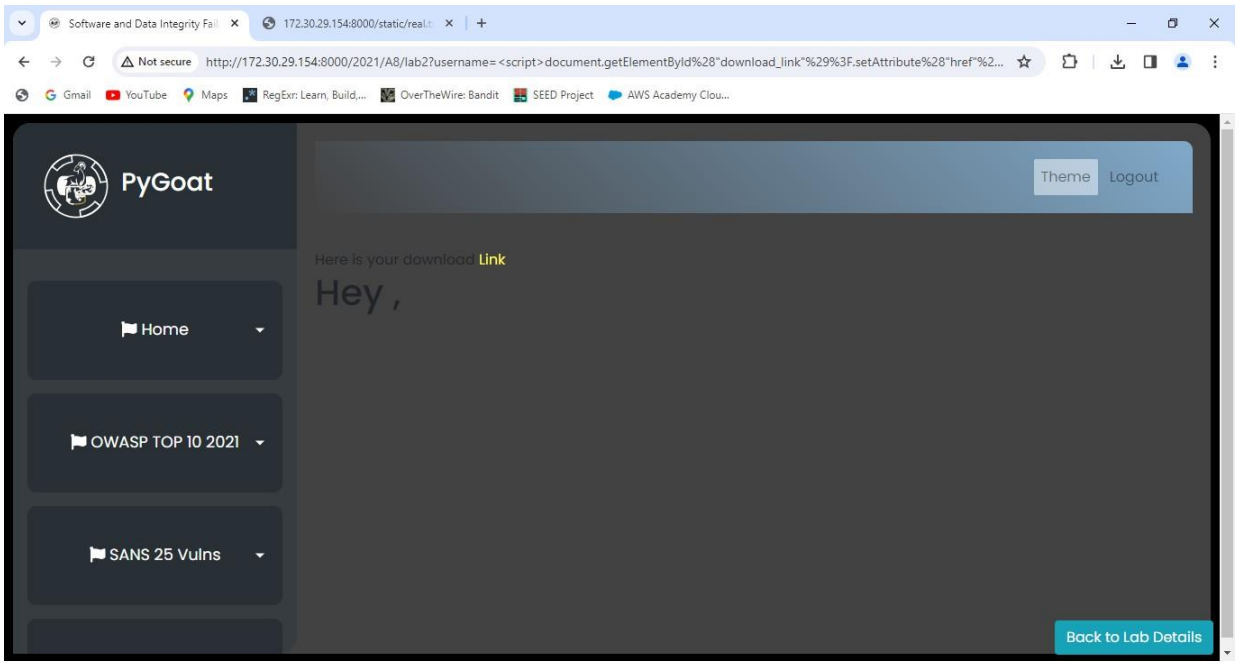
Với tên người dùng thông thường, ta thu được link tải file real.txt:

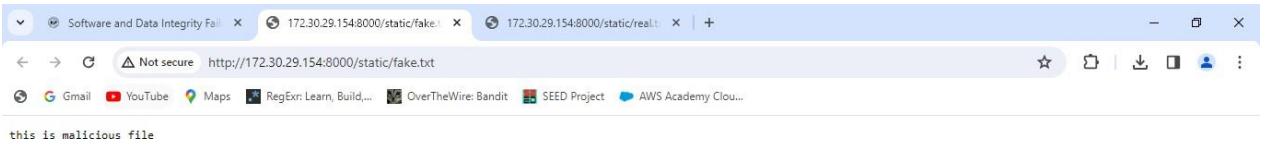


Thay vì dùng tên người dùng, ta thử dùng đoạn script để tại file fake.txt:

<script>document.getElementById("download_link")?.setAttribute("href", "/static/fake.txt");</script>


Software and Data Integrity Fail
172.30.29.154:8000/static/real
Not secure
http://172.30.29.154:8000/2021/A8/lab2
Gmail
YouTube
Maps
RegExr: Learn, Build,...
OverTheWire: Bandit
SEED Project
AWS Academy Clou...
PyGoat
Theme
Logout
Home
OWASP TOP 10 2021
SANS 25 Vulns
Your name ?
<script>document.getEler
Get download link
Back to Lab Details


Software and Data Integrity Fail
172.30.29.154:8000/static/real
Not secure
http://172.30.29.154:8000/2021/A8/lab2?username=<script>document.getElementById%28"download_link"%29%3F.setAttribute%28"href"%2...
Gmail
YouTube
Maps
RegExr: Learn, Build,...
OverTheWire: Bandit
SEED Project
AWS Academy Clou...
PyGoat
Theme
Logout
Here is your download Link
Hey ,
Home
OWASP TOP 10 2021
SANS 25 Vulns
Back to Lab Details



Mức độ ảnh hưởng: high

Tác động bảo mật: Kẻ tấn công có thể lợi dụng lỗ hổng này để thực hiện các cuộc tấn công như thay đổi dữ liệu, triển khai mã độc hại hoặc gây ra sự cố cho hệ thống bằng cách sửa đổi mã nguồn hoặc dữ liệu quan trọng.

Khuyến cáo khắc phục:

- Thực hiện các biện pháp kiểm tra tính toàn vẹn của dữ liệu sau khi nó đã được tải lên hệ thống. Sử dụng chữ ký số hoặc các cơ chế tương tự để xác minh phần mềm hoặc dữ liệu đến từ nguồn dự kiến và không bị thay đổi.

- Kiểm tra và xác thực mọi dữ liệu trước khi chấp nhận và xử lý, đảm bảo rằng chỉ dữ liệu hợp lệ mới được chấp nhận.

Tài liệu tham khảo:

https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures/

4. Kịch bản 04

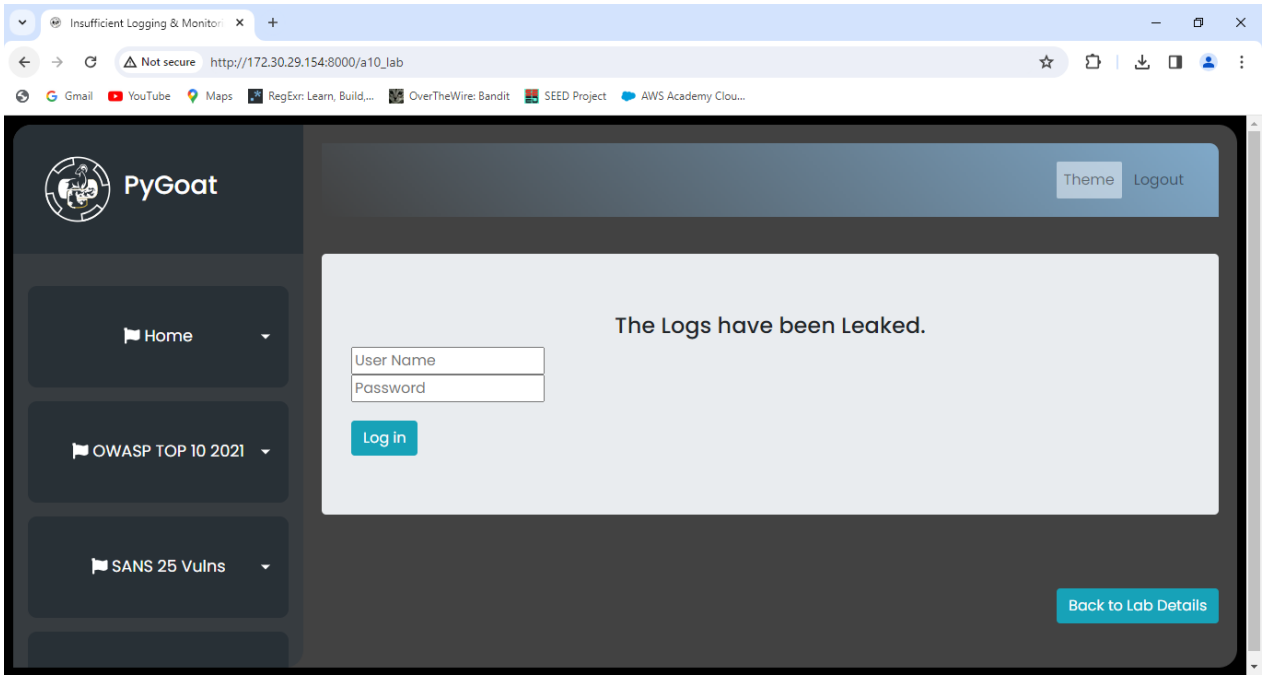
Tiêu đề: Security Logging and Monitoring Failures. Tài sản bị ảnh hưởng: thông tin, dữ liệu ghi trong log

Mô tả lỗ hổng:

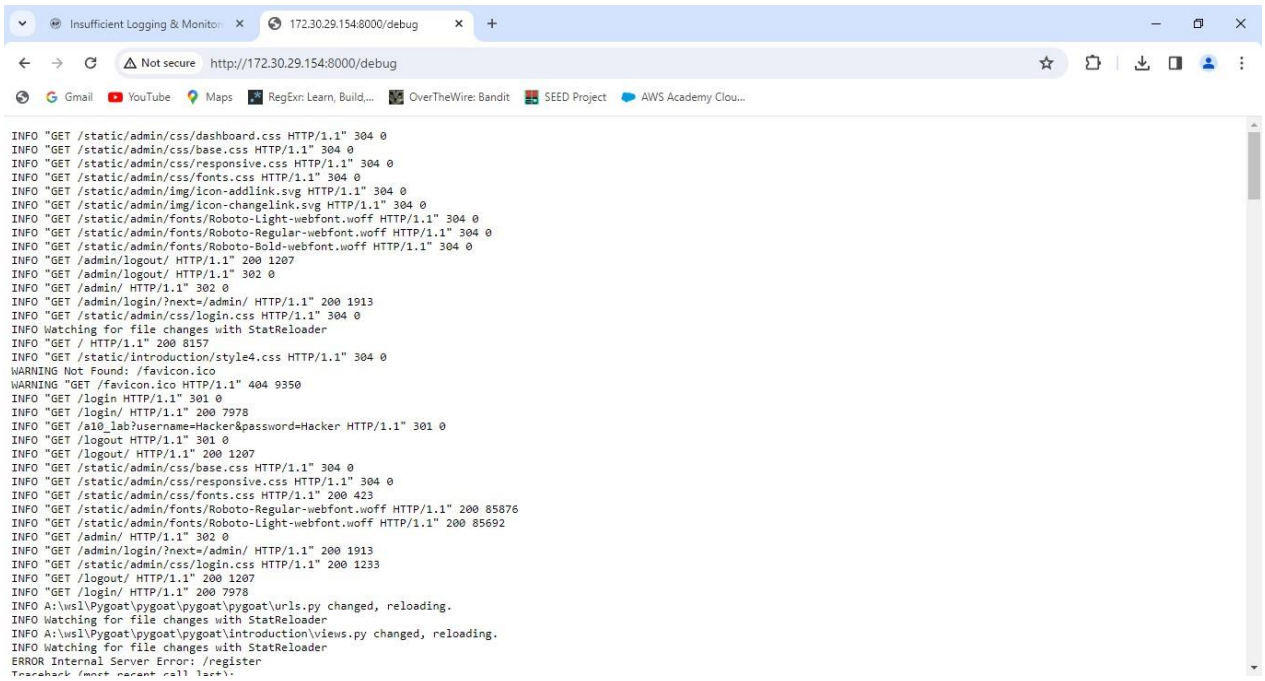
Lỗ hổng "Security Logging and Monitoring Failures" xảy ra khi hệ thống không thực hiện việc ghi log và giám sát hoạt động của người dùng một cách đầy đủ hoặc hiệu quả. Trong trường hợp này, kẻ tấn công có thể thực hiện các hành động tấn công trên các trang chỉ có quyền truy cập của admin mà không để lại dấu vết trong các hệ thống ghi log hoặc không bị phát hiện thông qua giám sát hoạt động.

Các bước thực hiện:

Đầu tiên ta truy cập vào http://localhost:8000/a10_lab



Từ gợi ý, ta truy cập vào route debug để lấy thông tin:




```
INFO "GET /login HTTP/1.1" 301 0
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
INFO "GET /logout/ HTTP/1.1" 200 1207
```

Ta thu được username = Hacker và password = Hacker

Mức độ ảnh hưởng: high

Tác động bảo mật nào mà kẻ tấn công có thể đạt được: Kẻ tấn công có thể tận dụng lỗ hổng này để thực hiện các cuộc tấn công mà không bị phát hiện hoặc bị truy kích. Họ có thể thực hiện các hành động độc hại như khai thác lỗ hổng bảo mật, đánh cắp dữ liệu nhạy cảm, ...

Khuyến cáo khắc phục:

- Đảm bảo các log được mã hóa chính xác để ngăn chặn việc tiêm nhiễm hoặc tấn công vào hệ thống ghi log hoặc giám sát.

- Sử dụng các biện pháp xác thực hai yếu tố (2FA) và hạn chế quyền truy cập theo nguyên tắc "tối thiểu quyền hạn".

Tài liệu tham khảo:

https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures/

5. Kịch bản 05

Tiêu đề: Server-Side Request Forgery (SSRF). Tài sản bị ảnh hưởng: các tập tin nội bộ của server.

Mô tả lỗ hổng:

Lỗ hổng "Server-Side Request Forgery (SSRF)" xảy ra khi kẻ tấn công có thể tạo ra các yêu cầu từ máy chủ đến các địa chỉ mà họ kiểm soát, thậm chí có thể là các hệ thống nội bộ hoặc không công khai. Trong tình huống này, kẻ tấn công thường sử dụng các yêu cầu này để khai thác hoặc tấn công các hệ thống khác trong mạng nội bộ hoặc bên ngoài.

Các bước thực hiện:

- Bước 1

Tìm các thành phần thực hiện các phương thức POST và GET.Như trong hình bên dưới là các button.

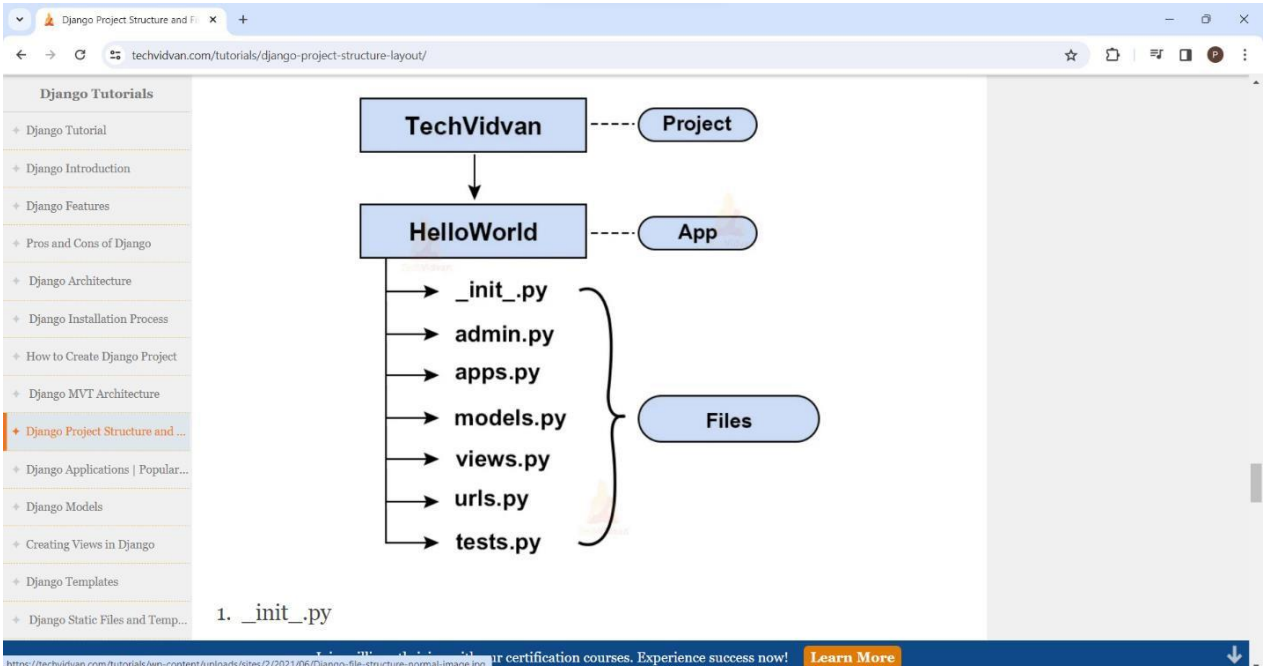
Như ta thấy ở hình dưới, trường values đọc dữ liệu từ các đường dẫn nội bộ và trả về giá trị trong đường dẫn đó.

- Bước 2

Từ code, ta thấy ngôn ngữ được sử dụng ở phía back-end là python Django.

```
def ssrf_lab(request):
 if request.user.is_authenticated:
  if request.method=="GET":
   return render(request,"Lab/ssrf/ssrf_lab.html",{"blog":"Read Blog About SSRF"})
  else:
   file=request.POST["blog"]
   try :
    dirname = os.path.dirname(__file__)
    filename = os.path.join(dirname, file)
    file = open(filename,"r")
    data = file.read()
    return render(request,"Lab/ssrf/ssrf_lab.html",{"blog":data})
   except:
    return render(request, "Lab/ssrf/ssrf_lab.html", {"blog": "No blog found"})
 else:
  return redirect('login')
```

Do đó, ta sẽ tìm hiểu cấu trúc của 1 web được code bằng Django.



- Bước 3

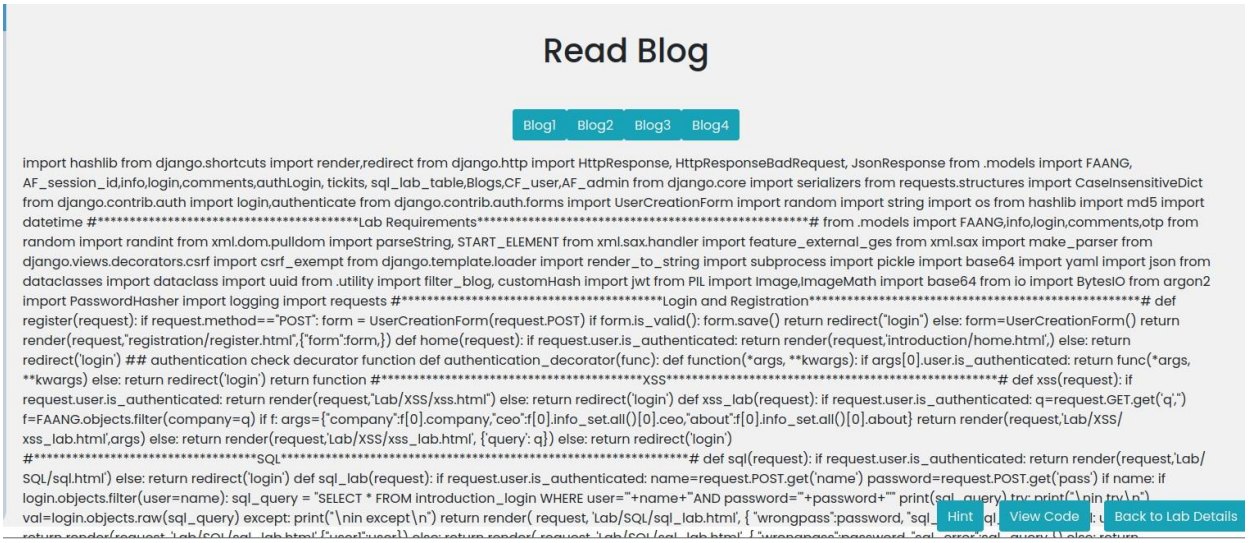
Tiến hành đổi đường dẫn thành các file phổ biến của framework này để lấy dữ liệu trả về của file đó.

Ví dụ ta sẽ đọc file views.py của trang web.

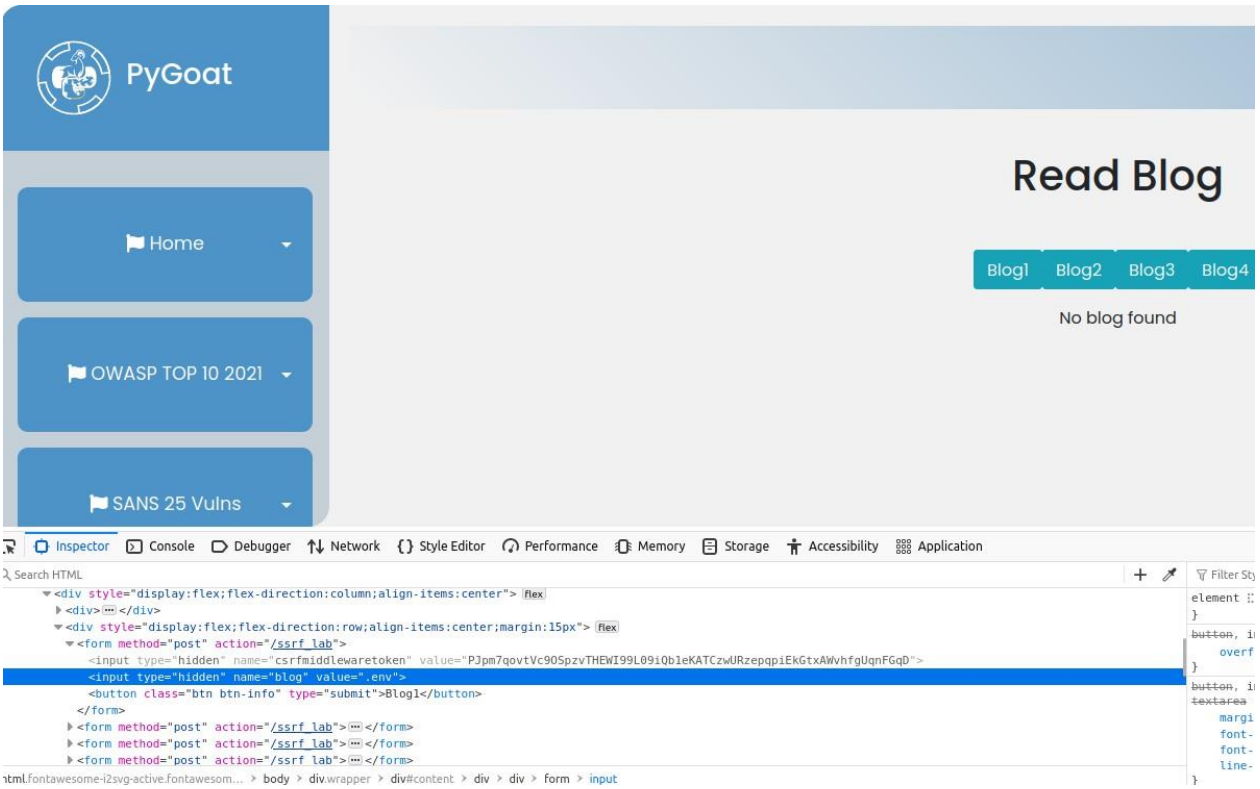
Đổi đường dẫn trong trường value thành views.py.

```
<form method="post" action="/ssrf_lab">
  <input type="hidden" name="csrfmiddlewaretoken" value="sbynUYYpJnRpgnKzXds9AZHkPAbbHRpMn22Dm4uLPG4GRNZk8Ni5AMPqkusgwmOb">
  <input type="hidden" name="blog" value="views.py">
  <button class="btn btn-info" type="submit">Blog1</button>
</form>
```

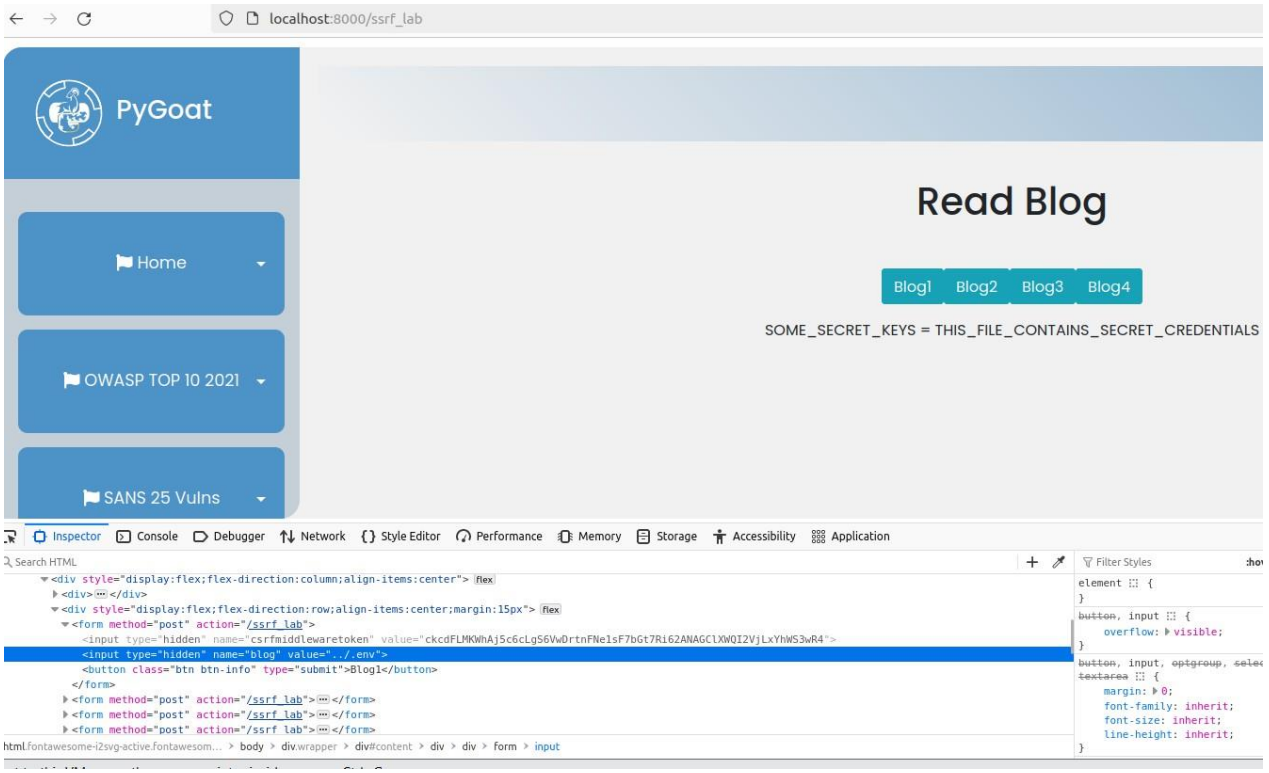
Sau khi nhấn vào button này , ta nhận được dữ liệu từ file views.py.



Tiếp theo, từ hint yêu cầu tìm file .env. Vì vậy ta tiến hành đổi đường dẫn trong trường value lần lượt là .env



Chưa có kết quả, tiếp tục ta tìm kiếm file .env trong các thư mục con bên dưới với ../.env thì thu được kết quả như hình bên dưới.



Mức độ ảnh hưởng: high

Tác động bảo mật nào mà kẻ tấn công có thể đạt được:

- Đọc dữ liệu từ máy chủ nội bộ: Kẻ tấn công có thể sử dụng SSRF để đọc dữ liệu từ máy chủ nội bộ, bao gồm các tệp cục bộ, thông tin kết nối và cấu hình hệ thống nội bộ. Điều này có thể tiết lộ thông tin nhạy cảm hoặc cung cấp thông tin cần thiết cho các cuộc tấn công tiếp theo.

- Tấn công truy cập vào dịch vụ nội bộ: Kẻ tấn công có thể sử dụng SSRF để tạo ra các yêu cầu HTTP hoặc các loại yêu cầu khác đến các dịch vụ nội bộ, chẳng hạn như giao thức FTP, SSH, Redis, hoặc MongoDB, có thể dẫn đến việc tấn công trực tiếp lên các dịch vụ này.

- Phạm vi của cuộc tấn công từ xa: SSRF có thể được sử dụng như một điểm vào mạng nội bộ từ xa, cho phép kẻ tấn công khai thác các lỗ hổng khác trong mạng nội bộ hoặc tạo ra các cuộc tấn công từ xa khác.

Khuyến cáo khắc phục:

- Xác thực và kiểm tra đầu vào: Đảm bảo rằng tất cả các URL được chấp nhận từ người dùng đều được kiểm tra kỹ lưỡng và chỉ chấp nhận các URL hợp lệ.

- Hạn chế quyền truy cập: Hạn chế quyền truy cập của ứng dụng đến các tài nguyên nội bộ và chỉ cho phép truy cập vào các tài nguyên cần thiết.

- Sử dụng whitelist thay vì blacklist: Thay vì chỉ định các tài nguyên không được phép truy cập, nên sử dụng whitelist để chỉ định các tài nguyên được phép truy cập.

Tài liệu tham khảo:

https://cheatsheetseries.owasp.org/cheatsheets/Server_Side_Request_Forgery_Prevention_Cheat_Sheet.html

---

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.
- 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).
- 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]-SessionX_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành đã đăng ký với GVHD-TH).

  *Ví dụ: [*NT101.K11.ATCL*]-Session1_Group3.*
- 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 Lab.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện được bài thực hành. Khuyến khích:

- Chuẩn bị tốt.
- Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây dựng bài thực hành.

Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.

HẾT