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

BÁO CÁO THỰC HÀNH

Môn học: Bảo mật Web và Ứng dụng

Tên chủ đề: Lab 2 – Tổng quan các lỗ hổng bảo mật Web thường gặp (P2)

GVHD: Ngô Đức Hoàng Sơn

1. **THÔNG TIN CHUNG:**

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT213.O21.ANTT.1

STT	Họ và tên	MSSV	Email
1	Vũ Tuấn Sơn	21521389	21521389@gm.uit.edu.vn
2	Bùi Đức Anh Tú	21522735	21522735@gm.uit.edu.vn
3	Lê Huy Hiệp	21522067	21522067@gm.uit.edu.vn

2. **NỘI DUNG THỰC HIỆN:**[1]

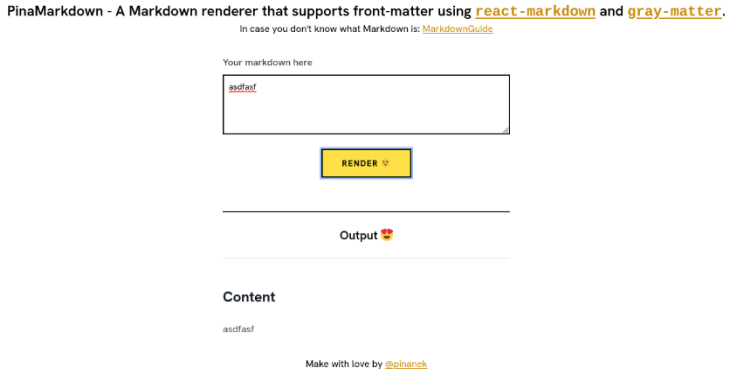
STT	Nội dung	Tình trạng	Trang
1			
2			
3	…	…	…
Điểm tự đánh giá			**?/10**

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

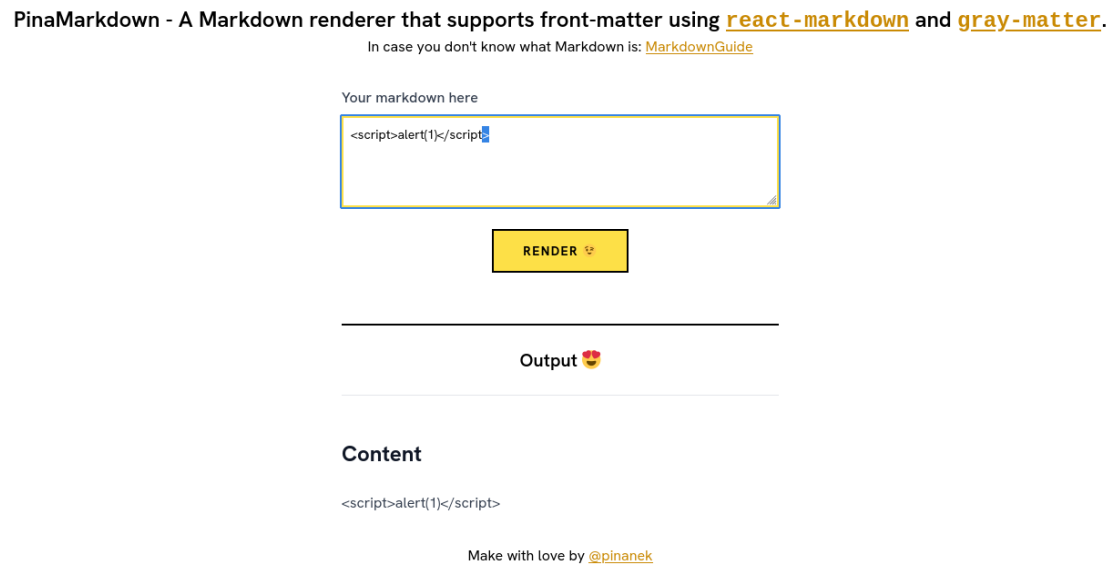
[1] Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

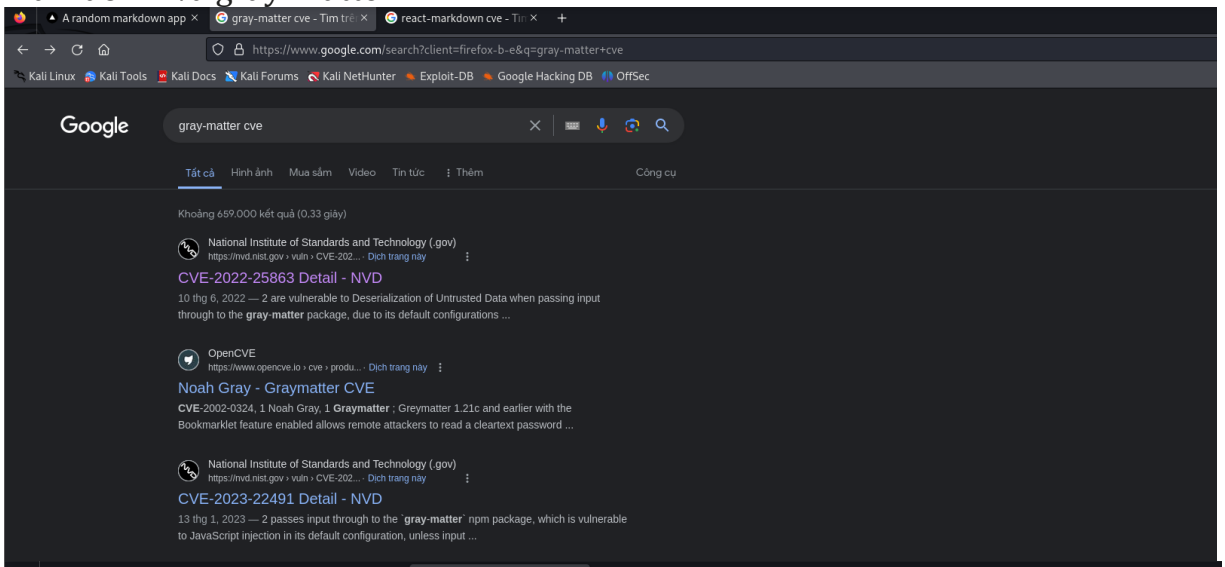
Bài thực hành số 1: Thực hiện việc khai thác lỗ hổng với một ứng dụng render Markdown thành HTML. Sử dụng format sau mẫu để trình bày.
#Tiêu đề: Khai thác lỗ hổng CVE-2022-25863 trong ứng dụng web render Markdown thành HTML.
#Mô tả lỗ hổng: Lỗ hổng CVE-2022-25863 liên quan đến gói gatsby-plugin-mdx củacác phiên bản trước 2.14.1, từ 3.0.0 và trước 3.15.2 của gói này đều có thể bị tấn công thông qua việc giải mã dữ liệu không đáng tin cậy. Lỗ hổng này xuất hiện khi dữ liệu đầu vào được chuyển qua gói gray-matter mà không được rà soát trước.
Việc khai thác lỗ hổng này có thể thực hiện được khi dữ liệu đầu vào được truyền vào cả hai chế độ: webpack (các tệp MDX trong src/pages hoặc tệp MDX được nhập như một thành phần trong mã nguồn frontend/React) và chế độ data (truy vấn các nút MDX qua GraphQL).
Các bước để thực hiện lại và bằng chứng:
Quan sát tổng quan ứng dụng, ta thấy giới thiệu về app sẽ có 1 form thực hiện render tất cả những gì nhập vào từ Markdown thành mã HTML.

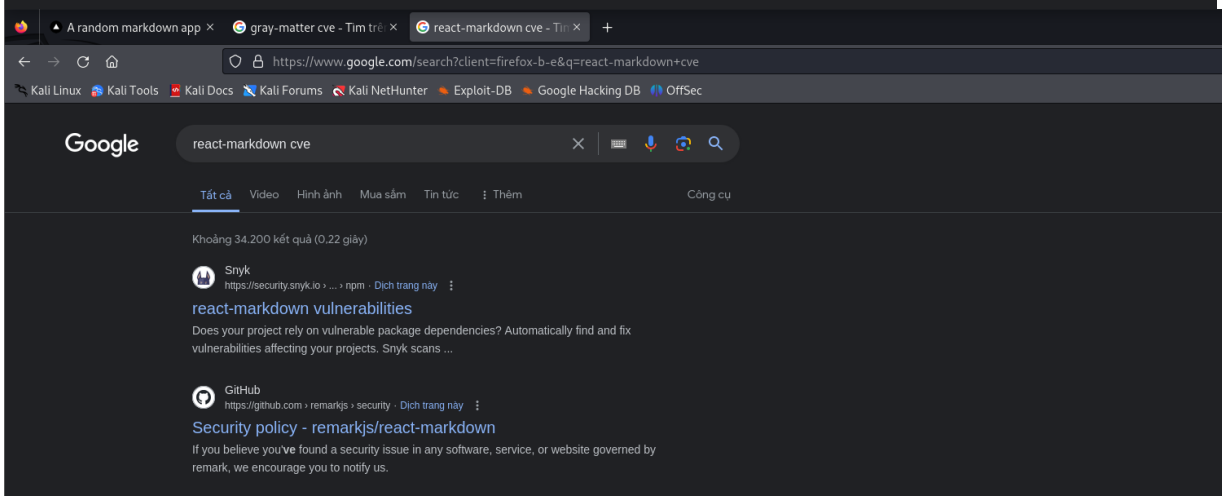


Ta sẽ nghĩ ngay đến việc thử XSS nó:

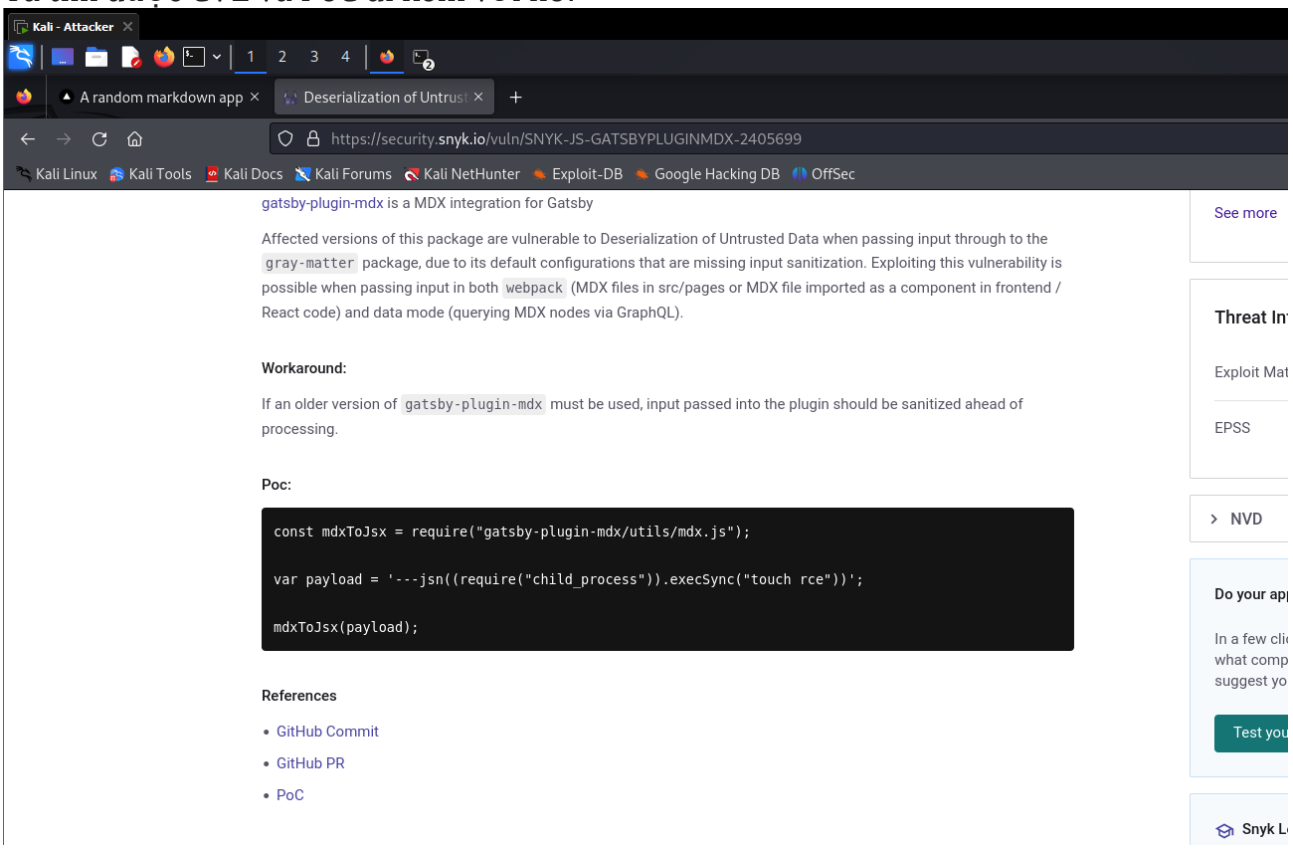


Không thành công. Chợt nhớ đến tiêu đề của bài tập này là về lỗ hổng bảo mật của những Components đã bị outdate. Như vậy nhiều khả năng web app này sẽ được tạo nên từ những Components như thế và có CVE về nó. Tìm kiếm về CVE của react-markdown và gray-matter:

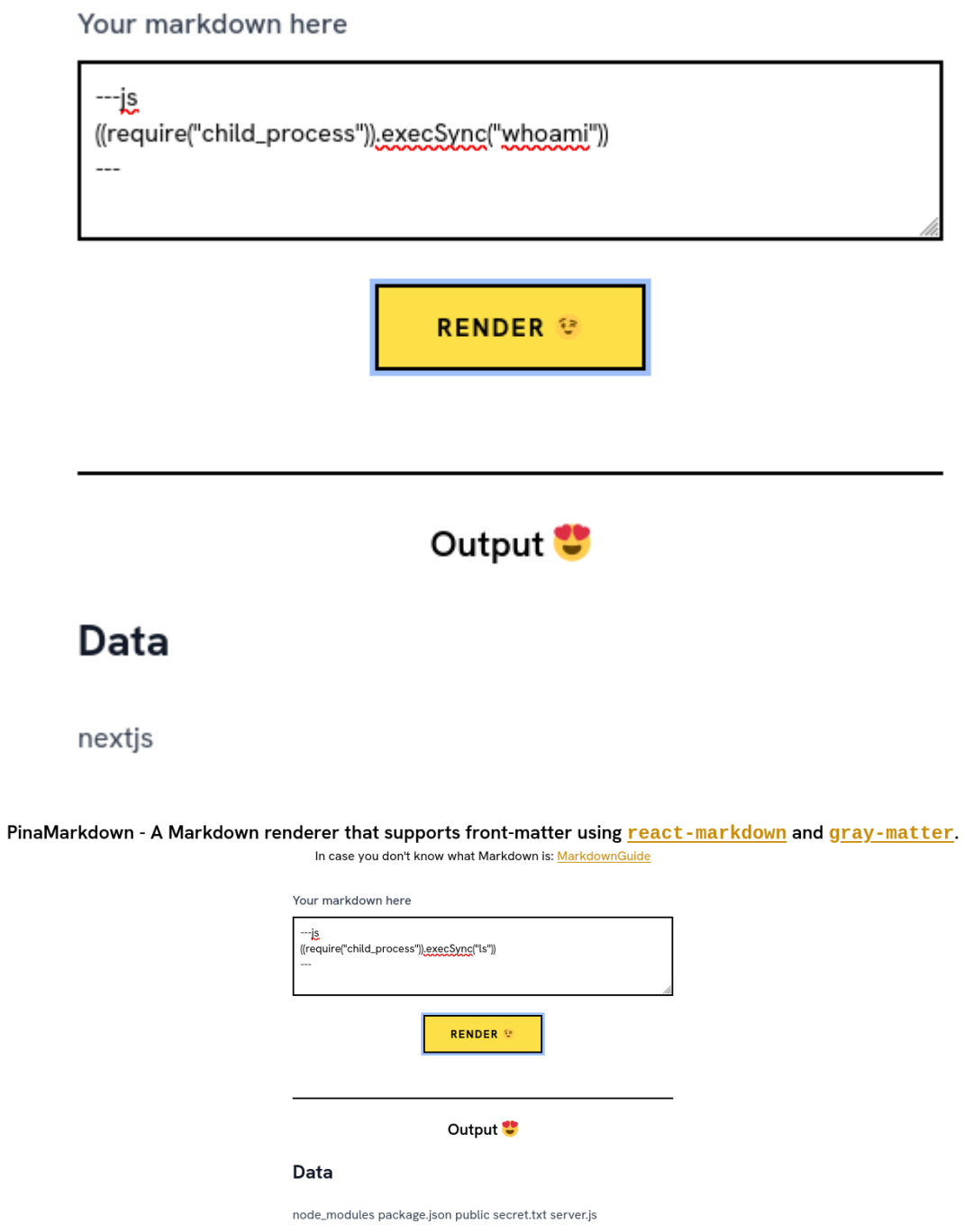




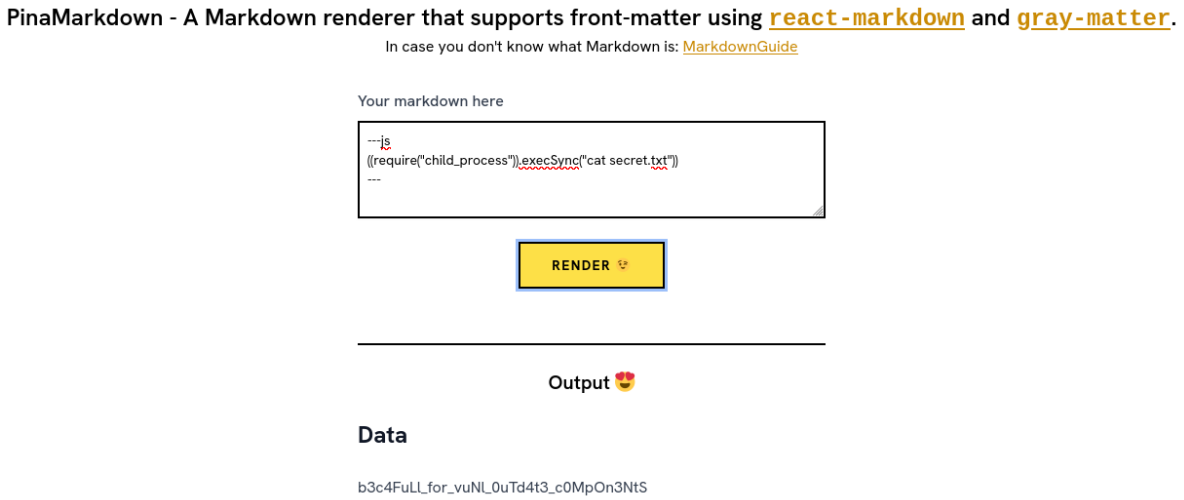
Ta tìm được CVE và PoC đi kèm với nó:



Ta nhận thấy từ đoạn code của CVE, Dòng thứ hai var payload = '---jsn((require("child_process")).execSync("touch rce"))'; khai báo một biến payload và gán giá trị là một chuỗi đại diện cho nội dung MDX. Thông qua chuỗi này, có sử dụng đoạn mã JavaScript require("child_process")).execSync("touch rce"). Đoạn mã này sử dụng module child_process trong Node.js để thực thi lệnh "touch rce", tức là tạo một file có tên là "rce". Như vậy ta hoàn toàn có thể thực hiện RCE với đoạn payload này. Thử 1 vài câu lệnh:

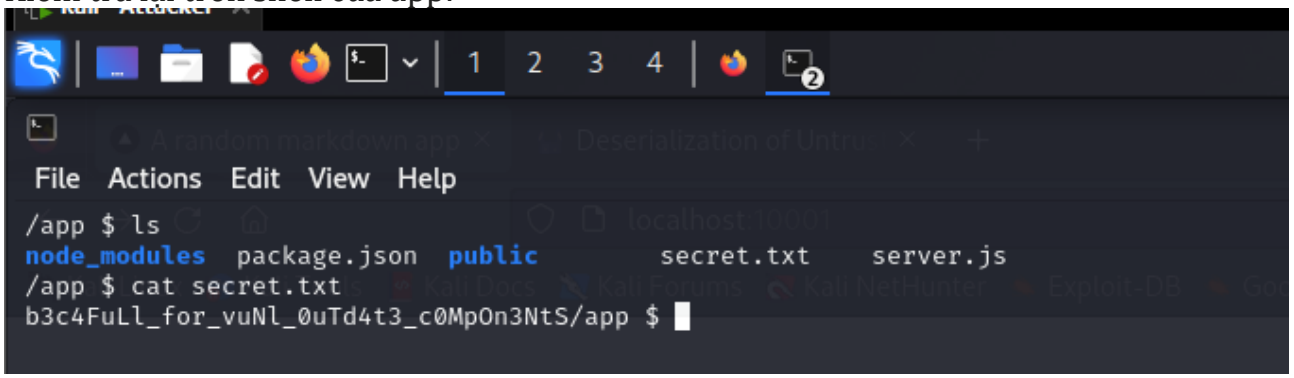


Có 1 file tên secret.txt, xem nội dung và nhận được flag cần tìm:



Như vậy flag cần tìm: b3c4FuLl_for_vuNl_0uTd4t3_c0MpOn3NtS

Kiểm tra lại trên shell của app:



Tài liệu hỗ trợ và tham khảo:

https://security.snyk.io/vuln/SNYK-JS-GATSBYPLUGINMDX-2405699
https://nvd.nist.gov/vuln/detail/CVE-2022-25863

#Mức độ ảnh hưởng của lỗ hổng: Rất nghiêm trọng. Attacker có thể tiến hành thực thi code từ xa để kiểm soát hoàn toàn hệ thống và lấy thông tin nhạy cảm.
#Khuyến cáo khắc phục: Update các Components lên phiên bản mới nhất. Nếu sử dụng các phiên bản thấp hơn thì cần phải kiểm tra lại dữ liệu đầu vào trước khi tiến hành xử lí.

Bài thực hành số 2:

#Tiêu đề: Identification and Authentication Failures, danh tính của người dùng
#Mô tả lỗ hổng:
- Lỗi nhận dạng và xác thực có thể xảy ra khi các chức năng liên quan đến danh tính, xác thực hoặc quản lý phiên của người dùng không được triển khai dúng cách hoặc không được bảo vệ đầy đủ.
- Kẻ tấn công có thể khai thác các lỗi nhận dạng và xác thực bằng cách xâm phạm mật khẩu, khoá, mã thông báo phiên hoặc khai thác các lỗi triển khai khác để giả định danh tính của người dùng khác, tạm thời hoặc vĩnh viễn.
Tóm tắt: Cố gắng vô hiệu hoá tài khoản admin không thể đăng nhập được nữa

Các bước để thực hiện lại và bằng chứng:

1. Bước 1

Password hash

$argon2id$v=19$m=65536,t=3,p=4$Ub40KHiEbH9I3Bsd4VHQDA$4zsIHDmAbejFJmaZq8a2yVIJdHvfylDlQ85w3YRLMSQ

ta thấy mật khẩu được băm bằng thuật toán argon2id là 1 hàm băm có độ an toàn cực cao nên không thể dò tìm mật khẩu

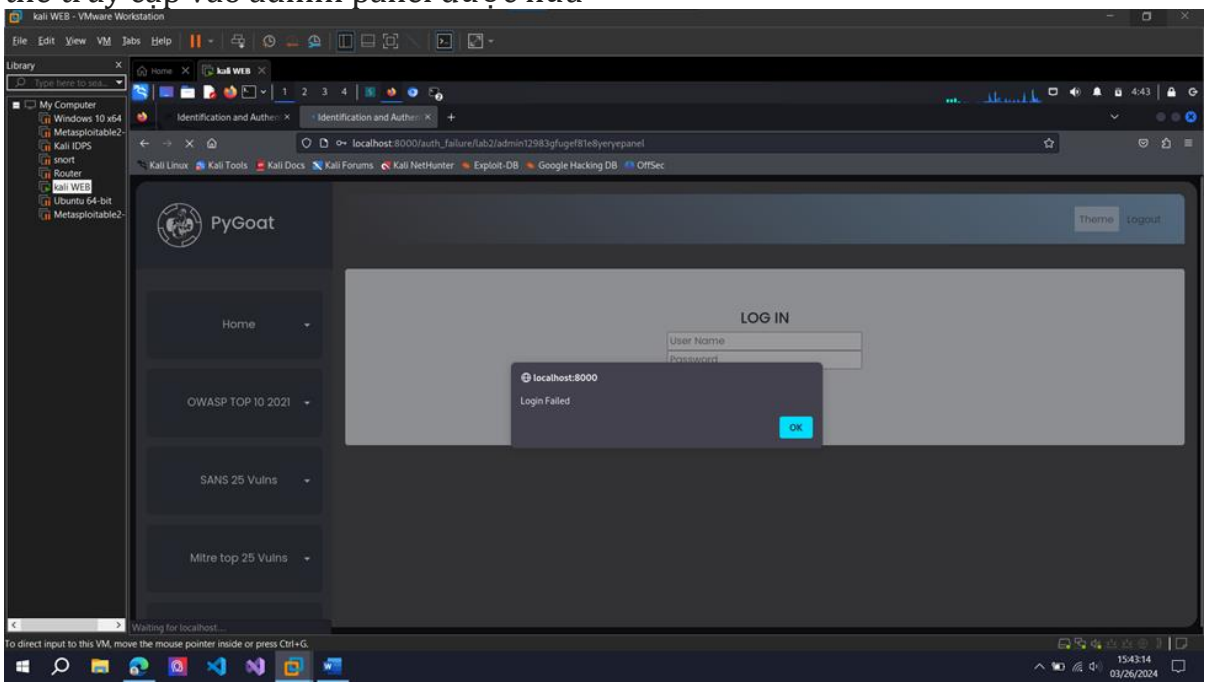
2. Bước 2

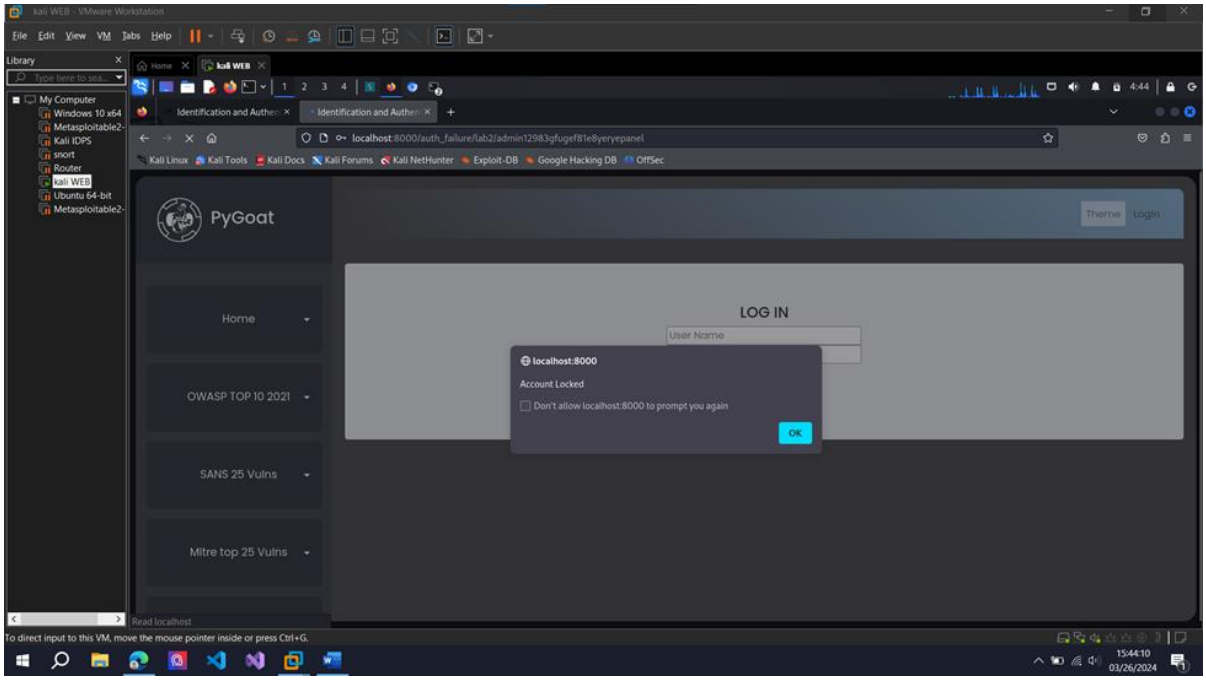
```
fail_attempt = user.failattempt + 1
            if fail_attempt == 5:
                user.is_active = False
                user.failattempt = 0
                user.is_locked = True
                user.lockout_cooldown = datetime.datetime.now() +
datetime.timedelta(minutes=1440)
                user.save()
```

Ta thấy ở trong đoạn code xử lý nếu chúng ta nhập sai mật khẩu của 1 tài khoản 5 lần thì tài khoản đó sẽ bị khoá 1440 phút(1 ngày)

3. Bước 3

Do đó chúng ta chỉ cần nhập sai tài khoản admin 5 lần thì quản trị viên thật sẽ không thể truy cập vào admin panel được nữa

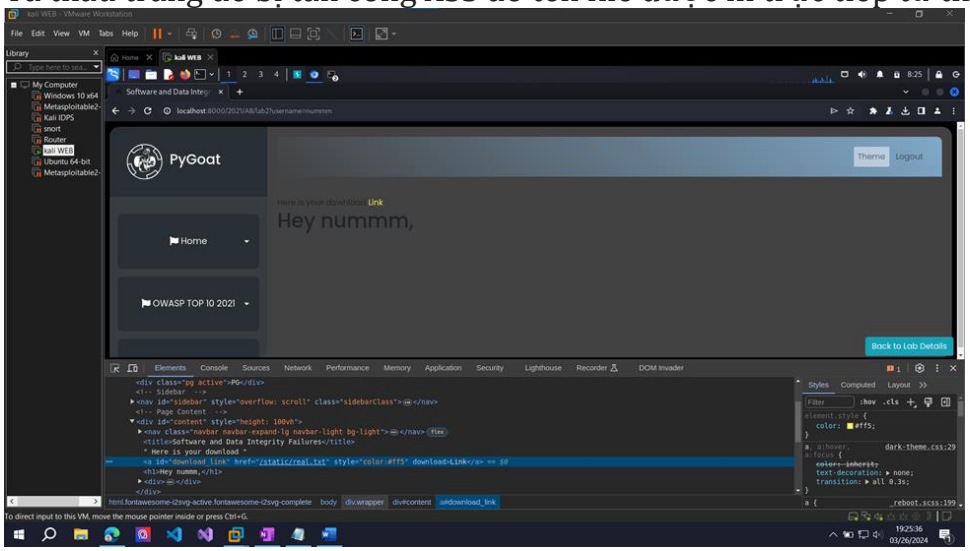




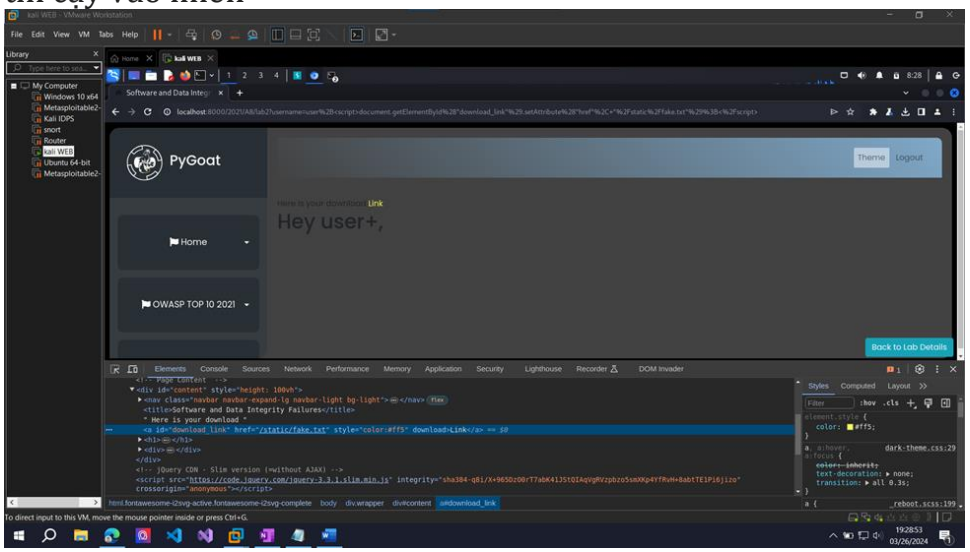
Tài liệu hỗ trợ và tham khảo:
A07 Identification and Authentication Failures - OWASP Top 10:2021
#Mức độ ảnh hưởng của lỗ hổng: Cao
- Kẻ tấn công có thể thực hiện các cuộc tấn công tự động như stuffing credentials, brute force, hoặc tái sử dụng thông tin đăng nhập bị đánh cắp
- Tiết lộ thông tin cá nhân, mất quyền kiểm soát tài khoản, và thậm chí là vi phạm dữ liệu trên diện rộng
#Khuyến cáo khắc phục:
- Triển khai xác thực đa yếu tố để ngăn chặn các cuộc tấn công tự động.
- Không sử dụng thông tin đăng nhập mặc định, đặc biệt là cho người dùng admin.
- Thực hiện kiểm tra mật khẩu yếu
- Đảm bảo rằng quy trình đăng ký, khôi phục thông tin đăng nhập, và đường dẫn API được bảo vệ chống lại các cuộc tấn công liệt kê tài khoản.
- Hạn chế hoặc tăng dần thời gian chờ sau mỗi lần đăng nhập thất bại.
- Ghi lại tất cả các lần đăng nhập thất bại và thông báo cho quản trị viên khi phát hiện các cuộc tấn công

Bài thực hành số 3:
#Tiêu đề: A08:2021 – Software and Data Integrity Failures , tính toàn vẹn của dữ liệu
#Mô tả lỗ hổng: Các lỗi về tính toàn vẹn của phần mềm và dữ liệu liên quan đến mã và cơ sở hạ tầng không bảo vệ chống lại các vi phạm về tính toàn vẹn.
- Điều này có thể xảy ra khi sử dụng phần mềm từ các nguồn và kho lưu trữ không đáng tin cậy hoặc thậm chí là phần mềm bị can thiệp tại nguồn, trong quá trình chuyển tiếp hoặc thậm chí là trong bộ đệm của endpoint. Kịch bản bài thực hành bao gồm một trang chức năng là hiển thị trang download tài liệu cho người dùng. Cố gắng thay đổi tính toàn vẹn của file

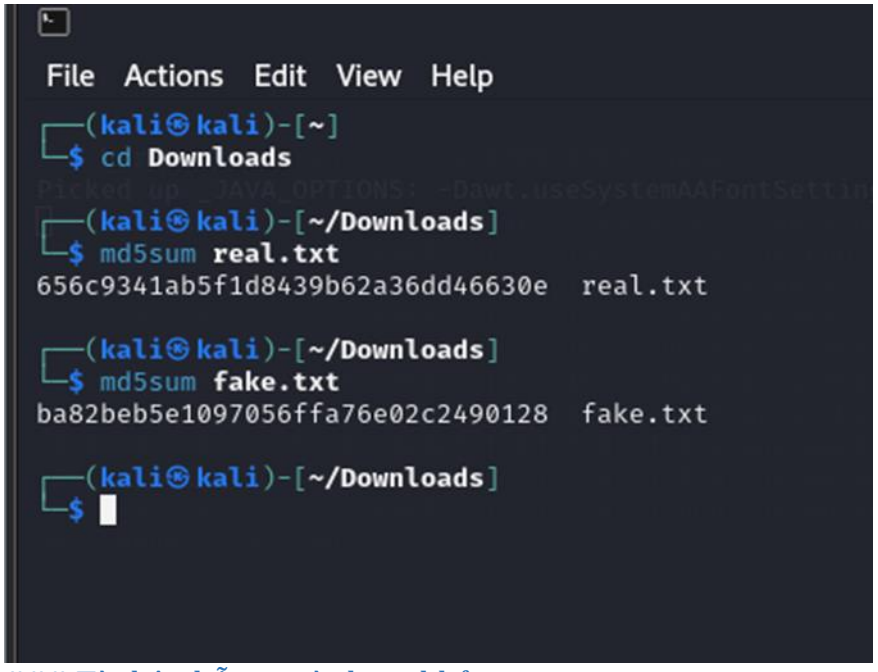
Tóm tắt: Thay đổi tính toàn vẹn của file download
Các bước để thực hiện lại và bằng chứng:
1. Bước 1
Ta thấu trang dễ bị tấn công XSS do tên file được in trực tiếp từ thông số url



2. Bước 2
Tiến hành tấn công XSS vào trang với input
user+<script>document.getElementById("download_link").setAttribute("href", "/static/fake.txt");</script>
3. Bước 3
Url tải xuống đã được sửa đổi và người dùng có thể tải xuống một tệp tùy ý bằng cách tin cậy vào miền



Kiểm tra 2 file thấy không giống nhau



Tài liệu hỗ trợ và tham khảo:
A08 Software and Data Integrity Failures - OWASP Top 10:2021
#Mức độ ảnh hưởng của lỗ hổng: Cao
- Kẻ tấn công có thể thay đổi hoặc thêm mã độc vào phần mềm hoặc dữ liệu mà không bị phát hiện. - Điều này có thể dẫn đến việc phân phối mã độc và tấn công vào tất cả các hệ thống sử dụng phần mềm hoặc dữ liệu bị ảnh hưởng
#Khuyến cáo khắc phục:
- Sử dụng chữ ký số hoặc các cơ chế tương tự để xác minh phần mềm hoặc dữ liệu là từ nguồn dự kiến và không bị thay đổi.
- Đảm bảo các thư viện và phụ thuộc, chẳng hạn như npm hoặc Maven, là tiêu thụ kho lưu trữ đáng tin cậy. Nếu bạn có hồ sơ rủi ro cao hơn, hãy cân nhắc lưu trữ một kho lưu trữ nội bộ đã biết tốt đã được kiểm tra.
- Đảm bảo rằng một công cụ bảo mật chuỗi cung ứng phần mềm, chẳng hạn như OWASP Kiểm tra phụ thuộc hoặc OWASP CycloneDX, được sử dụng để xác minh rằng Các thành phần không chứa lỗ hổng đã biết
- Đảm bảo rằng có quy trình xem xét các thay đổi về mã và cấu hình để giảm thiểu khả năng mã độc hoặc cấu hình có thể được đưa vào quy trình phần mềm của bạn.
- Đảm bảo rằng quy trình CI/CD của bạn có sự phân tách, cấu hình và truy cập thích hợp kiểm soát để đảm bảo tính toàn vẹn của mã chảy qua Xây dựng và triển khai các quy trình.
- Đảm bảo rằng dữ liệu tuần tự hóa không được ký hoặc không được mã hóa không được gửi đến Khách hàng không đáng tin cậy mà không có một số hình thức kiểm tra tính toàn vẹn hoặc kỹ thuật số chữ ký để phát hiện giả mạo hoặc phát lại dữ liệu được nối tiếp hóa

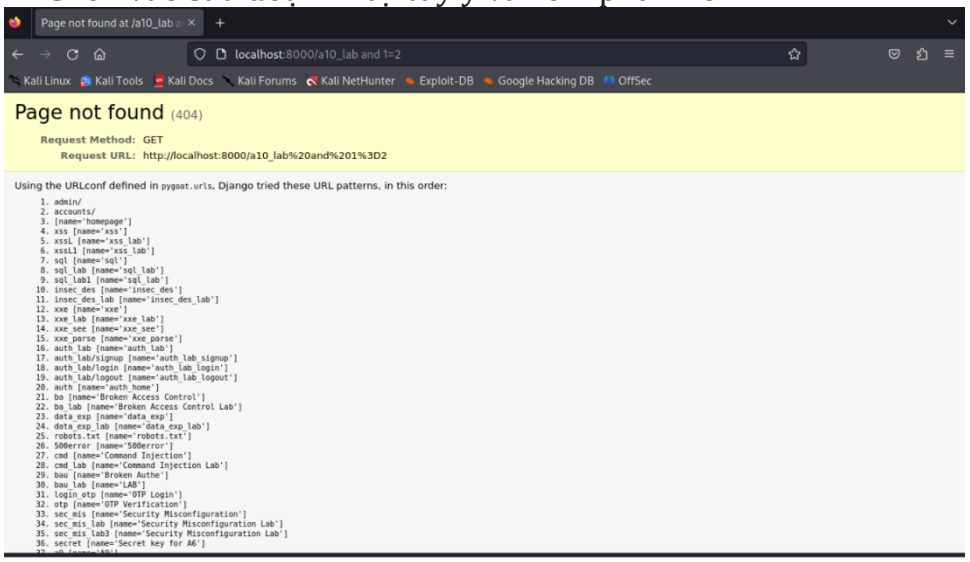
Bài thực hành số 4:

#Tiêu đề: A09:2021 – Security Logging and Monitoring Failures, tài sản ảnh hưởng hệ thống thông tin, dữ liệu quan trọng
#Mô tả lỗ hổng:
Lỗ hổng A09:2021 – Security Logging and Monitoring Failures bao gồm:

- Không ghi lại đầy đủ: Hệ thống không ghi lại đầy đủ thông tin về các sự kiện bảo mật quan trọng như đăng nhập không thành công, truy cập vào các tài nguyên nhạy cảm, hoặc các hoạt động không bình thường.
- Không giám sát được: Hệ thống không thể giám sát hoặc phát hiện các hoạt động xâm nhập hoặc không bình thường, dẫn đến việc không nhận biết được các mối đe dọa đang tiềm ẩn.
- Lỗi trong quá trình ghi lại và giám sát: Các công cụ hoặc quy trình liên quan đến ghi lại và giám sát có thể gặp phải các lỗi hoặc vấn đề kỹ thuật, dẫn đến việc mất mát dữ liệu hoặc thông tin không chính xác.
- Không tuân thủ chính sách và quy định: Hệ thống không tuân thủ các chính sách và quy định liên quan đến ghi lại và giám sát bảo mật, gây ra việc thiếu bảo vệ và kiểm soát.
- Không có quy trình giải quyết sự cố: Thiếu các quy trình hoặc phương tiện để xử lý các sự cố liên quan đến ghi lại và giám sát, dẫn đến việc không khẩn trương hoặc không hiệu quả trong việc phản ứng lại các vấn đề bảo mật.

Tóm tắt: Tìm kiếm bản ghi của trang web bị rò rỉ. Nó có thể chứa những thông tin hữu ích

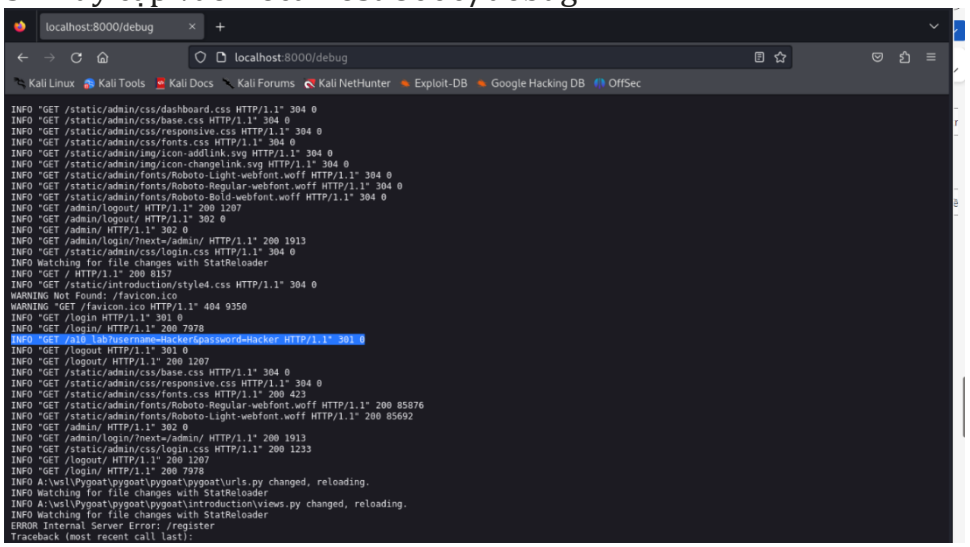
Các bước để thực hiện lại và bằng chứng:
1. Chèn vào sau đoạn kí tự tùy ý và xem phản hồi



Trang web bị lỗi và trả về những url hợp lệ được định nghĩa trong tệp pygoat.urls
2. Dòng số 44 debug có thể là log bị lộ

```
34. sec_mis_lab [name='Security Misconfiguration Lab']
35. sec_mis_lab3 [name='Security Misconfiguration Lab']
36. secret [name='Secret key for A6']
37. a9 [name='A9']
38. a9_lab [name='A9 LAb']
39. a9_lab2 [name='A9 LAb 2']
40. get_version [name='Get Version']
41. a10 [name='A10']
42. a10_lab [name='A10 LAb']
43. a10_lab_2 [name='A10 LAb 2']
44. debug [name='debug']
45. insecure-design [name='insecure-design']
46. insecure-design_lab [name='insecure-design_lab']
47. broken_access_control [name='broken_access']
48. broken_access_lab_1 [name='broken_access_lab_1']
49. broken_access_lab_2 [name='broken_access_lab_2']
50. broken_access_lab_3 [name='broken_access_lab_3']
51. broken_access_controle/secret [name='broken_access_controle_secret']
52. ssrf [name='SSRF']
```

3. Truy cập vào "localhost:8000/debug"



Thấy có 1 dòng yêu cầu đăng nhập vào "/a10_lab" với:
username=hacker và password=hacker

4. Thử đăng nhập

Tài liệu hỗ trợ và tham khảo:

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

#Mức độ ảnh hưởng của lỗ hổng: rất nghiêm trọng. Kẻ tấn công có thể: Xâm nhập dữ liệu, ẩn mình và theo dõi, phá hoại hệ thống,...

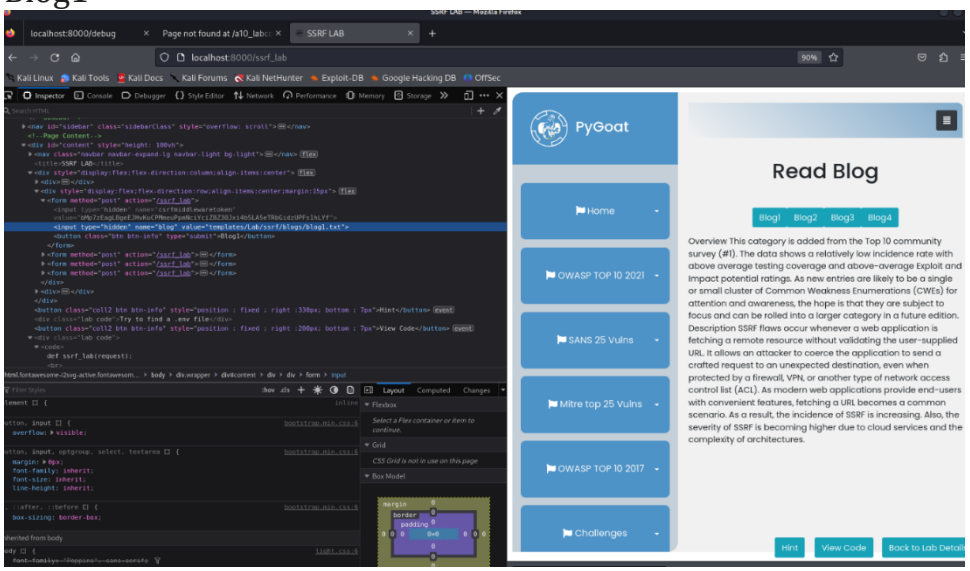
#Khuyến cáo khắc phục:

Nhà phát triển nên triển khai một số hoặc tất cả các biện pháp kiểm soát sau, tùy thuộc vào rủi ro của ứng dụng:

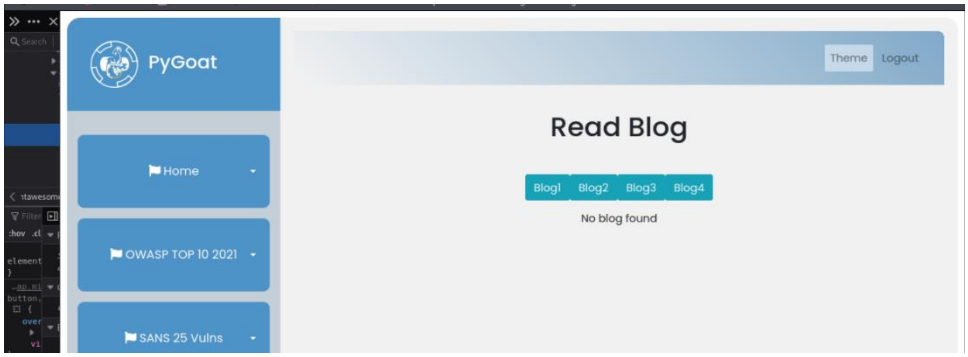
- Đảm bảo tất cả các lỗi đăng nhập, kiểm soát quyền truy cập và xác thực đầu vào phía máy chủ có thể được ghi lại với đầy đủ ngữ cảnh của người dùng để xác định các tài khoản đáng ngờ hoặc độc hại và được giữ trong thời gian đủ để cho phép trì hoãn phân tích điều tra.
- Đảm bảo rằng nhật ký được tạo ở định dạng mà các giải pháp quản lý nhật ký có thể dễ dàng sử dụng.
- Đảm bảo dữ liệu nhật ký được mã hóa chính xác để ngăn chặn việc tiêm nhiễm hoặc tấn công vào hệ thống ghi nhật ký hoặc giám sát.
- Đảm bảo các giao dịch có giá trị cao có quy trình kiểm tra với các biện pháp kiểm soát tính toàn vẹn để ngăn chặn việc giả mạo hoặc xóa, chẳng hạn như các bảng cơ sở dữ liệu chỉ nối thêm hoặc tương tự.

- Các nhóm DevSecOps nên thiết lập hoạt động giám sát và cảnh báo hiệu quả để phát hiện và phản hồi nhanh chóng các hoạt động đáng ngờ.
- Thiết lập hoặc thông qua kế hoạch ứng phó và khắc phục sự cố, chẳng hạn như Viện Tiêu chuẩn và Công nghệ Quốc gia (NIST) 800-61r2 trở lên.

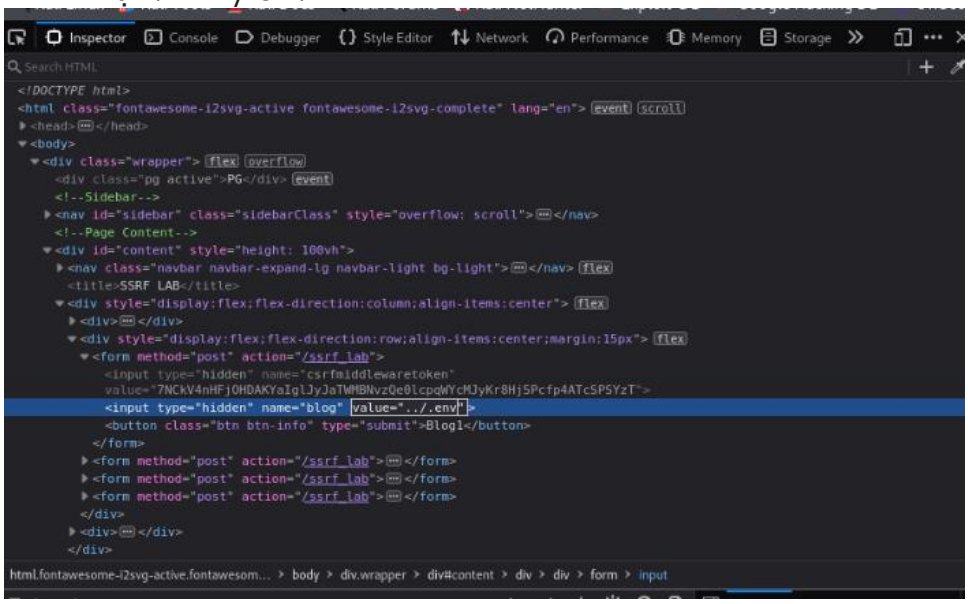
Bài thực hành số 5:
#Tiêu đề: A10:2021 – Server-Side Request Forgery (SSRF) tài sản bị ảnh hưởng là password, thông tin xác thực API,...
#Mô tả lỗ hổng: Lỗi SSRF xảy ra bất cứ khi nào ứng dụng web tìm nạp tài nguyên từ xa mà không xác thực URL do người dùng cung cấp. Nó cho phép kẻ tấn công ép buộc ứng dụng gửi yêu cầu được tạo ra đến một đích không mong muốn, ngay cả khi được bảo vệ bởi tường lửa, VPN hoặc một loại danh sách kiểm soát truy cập mạng (ACL) khác.
Tóm tắt: Thực hiện kiểm tra mã nguồn để tìm kiếm xem đường dẫn để lấy tài nguyên từ xa nằm ở đâu. Xem xét và chỉnh sửa đường dẫn để truy cập vào tài nguyên từ xa.
Các bước để thực hiện lại và bằng chứng:
1. Xem mã nguồn của trang web
2. Thường thì các button sẽ lấy hoặc gửi dữ liệu từ đâu đó. Xem mã nguồn của button Blog1



Thấy input được ấy từ đường dẫn “templates/Lab/ssrf/blogs/blog1.txt”
3. Thay đổi đường dẫn thành “.env”
Kết quả chưa tìm ra



4. Thử lại với "../.env"



Kết quả:

Tập tin ".env": Là tập tin dùng để lưu trữ các biến môi trường trong Node được sử dụng để lưu trữ dữ liệu nhạy cảm như mật khẩu, thông tin xác thực API và các thông tin khác không được ghi trực tiếp bằng mã. có chứa các cặp khóa/giá trị xác định các biến môi trường bắt buộc của dự án.

Tài liệu hỗ trợ và tham khảo:

https://www.codementor.io/@parthibakumarmurugesan/what-is-env-how-to-set-up-and-run-a-env-file-in-node-1pnyxw9yxj

https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29/

#Mức độ ảnh hưởng của lỗ hổng: Rất nghiêm trọng và có thể ảnh hưởng đến nhiều phía khác nhau như: Server bị tấn công; tấn công các thực thể bên ngoài, bao gồm các dịch vụ web hoặc hệ thống ngoại vi khác; xâm nhập dữ liệu, phân rã hệ thống.

#Khuyến cáo khắc phục: Các nhà phát triển có thể ngăn chặn SSRF bằng cách triển khai một số hoặc tất cả các biện pháp phòng thủ theo chiều sâu sau đây:

- Từ lớp Mạng:

- Phân chia chức năng truy cập tài nguyên từ xa vào các mạng riêng biệt để giảm thiểu tác động của SSRF.
- Thực thi các chính sách tường lửa hoặc quy tắc kiểm soát truy cập mạng " deny by default " để chặn tất cả lưu lượng mạng ngoại trừ lưu lượng mạng nội bộ cần thiết. Gợi ý:
- Thiết lập quyền sở hữu và vòng đời cho các quy tắc tường lửa dựa trên ứng dụng.
- Ghi lại tất cả các luồng mạng được chấp nhận và bị chặn trên tường lửa (xem A09:2021- Ghi lại và Giám sát Lỗi Bảo mật).

- Từ lớp Ứng dụng:

- Làm sạch và xác thực tất cả dữ liệu nhập từ người dùng.
- Bắt buộc các URL schema, cổng và điểm đích với một danh sách cho phép tích cực.
- Không gửi phản hồi nguyên thô đến người dùng.
- Vô hiệu hóa chuyển hướng HTTP.
- Chú ý đến tính nhất quán của URL để tránh các cuộc tấn công như DNS rebinding và "time of check, time of use" (TOCTOU).