

BÁO CÁO THỰC HÀNH

Môn học: Bảo mật web và ứng dụng

Tên chủ đề: Tổng quan về các lỗ hổng bảo mật web thường gặp (phần 2)

GVHD: Ngô Đức Hoàng Sơn

Nhóm: 12

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	Nguyễn Triệu Thiên Bảo	21520155	21520155@gm.uit.edu.vn
2	Trần Lê Minh Ngọc	21521195	21521195@gm.uit.edu.vn
3	Huỳnh Minh Khuê	21522240	21522240@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:¹

STT	Nội dung	Tình trạng	Trang
1	Yêu cầu 1	100%	2 - 6
2	Yêu cầu 2	100%	7 - 9
3	Yêu cầu 3	100%	9 - 11
4	Yêu cầu 4	100%	11 - 13
5	Yêu cầu 5	100%	13 - 15
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

a) A06:2021 – Vulnerable and Outdated Components

Lỗi bảo mật OWASP Top 10 Vulnerable and Outdated Components

Tiêu đề: Vulnerable and Outdated Components - Lỗ hổng liên quan tới các ứng dụng lỗi thời và chứa lỗ hổng nhưng không được cập nhật hoặc vá lỗi kịp thời.

Mô tả lỗ hổng:

- **Tóm tắt:** Trang web cho phép chuyển đổi yaml sang json. Tuy nhiên trang web sử dụng pyyaml 5.1, một phiên bản đã cũ và có lỗi liên quan tới thực thi code khi upload file.
- **Các bước để thực hiện lại và bằng chứng:**

Bước 1: Ta tạo 1 file python chứa class bất kì. Trong class này ta sẽ khai báo phương thức `__reduce__`. Phương thức giúp pyyaml biết làm sao để xử lý một loại dữ liệu nào đó, ở đây là class do ta tự định nghĩa.

```
(kali@kali)-[~/Downloads]
$ cat cau1.py
import yaml
import subprocess

class Payload(object):
    def __reduce__(self):
        return (subprocess.Popen, ('ls',))

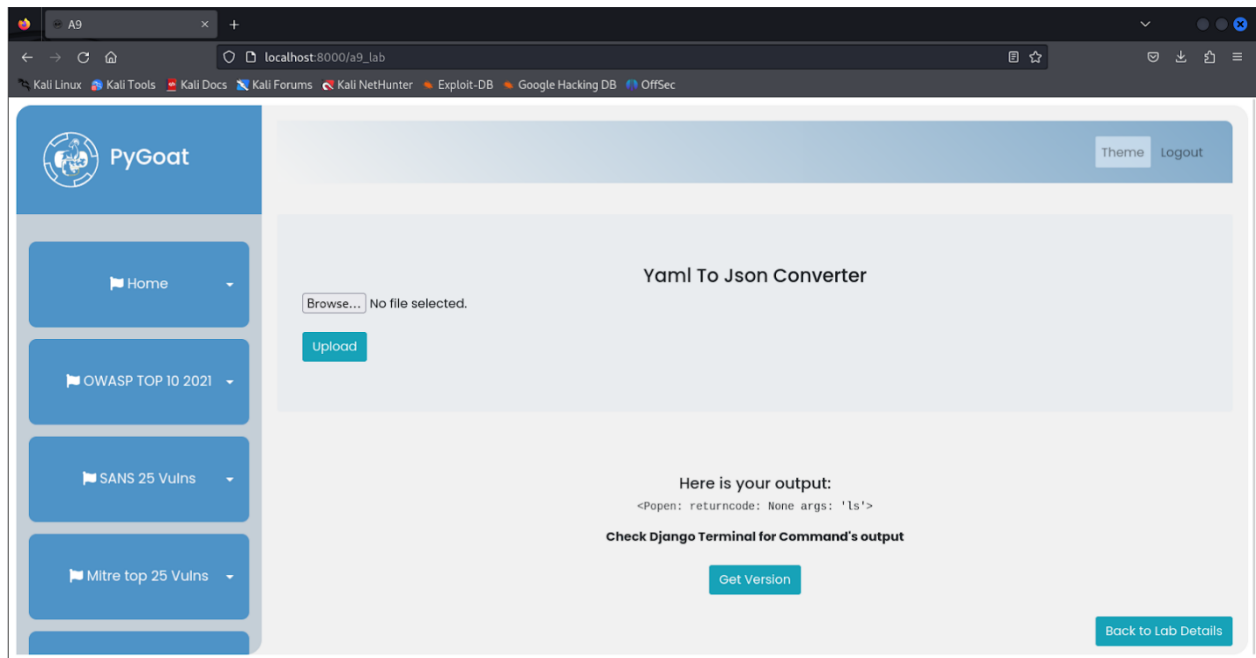
deserialized_data = yaml.dump(Payload())
print(deserialized_data)

(kali@kali)-[~/Downloads]
$
```

Bước 2: Ở phương thức `__reduce__`, ta sẽ thực hiện trả về 1 tuple với phần tử thứ nhất là hàm ta cần gọi, ở đây là `subprocess.Popen`, và phần tử tiếp theo là tham số. Thực hiện đoạn code và có được payload:

```
(kali@kali)-[~/Downloads]
$ python cau1.py
!! python/object/apply:subprocess.Popen
- ls
```

Bước 3: Tạo file yaml với nội dung như trên và upload file lên trang web. Ta có kết quả:



Mức độ ảnh hưởng của lỗ hổng: Cao. Lỗ hổng này có thể gây ra tác động nghiêm trọng đến bảo mật của hệ thống. Các thành phần phần mềm cũ, không được cập nhật đều đặn hoặc chứa các lỗ hổng bảo mật có thể tạo ra các điểm yếu cho kẻ tấn công tấn công vào hệ thống một cách dễ dàng.

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.

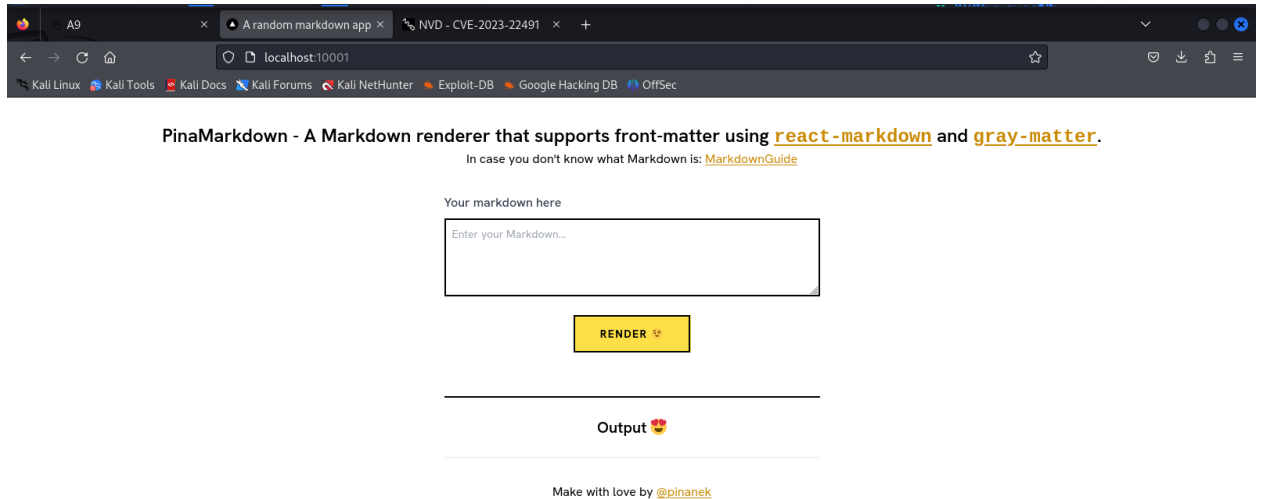
Khuyến cáo khắc phục: Cập nhật phần mềm thường xuyên. Có filter khi upload file lên server, chặn quyền thực thi của file

Bài tập 1: Thực hiện việc khai thác lỗ hổng với một ứng dụng render Markdown thành HTML.

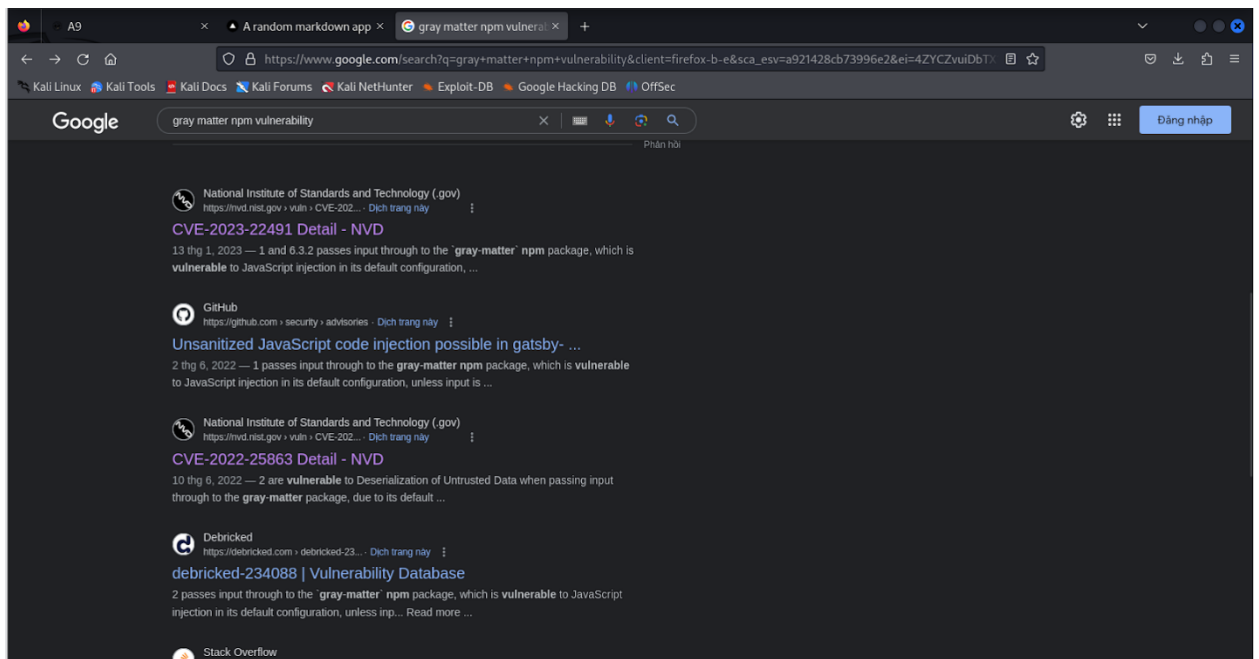
Tóm tắt: Trang web cho phép render Markdown thành HTML. Tuy nhiên trang web sử dụng grey-matter phiên bản cũ và có lỗi liên quan tới thực thi code khi input.

Các bước để thực hiện lại và bằng chứng:

Bước 1: Khi vào trang web ta thấy trang web giới thiệu được viết bằng react-markdown và gray-matter.



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



NATIONAL VULNERABILITY DATABASE

NOTICE

NIST is currently working to establish a consortium to address challenges in the NVD program and develop improved tools and methods. You will temporarily see delays in analysis efforts during this transition. We apologize for the inconvenience and ask for your patience as we work to improve the NVD program.

CVE-2023-22491 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Current Description

Gatsby is a free and open source framework based on React that helps developers build websites and apps. The gatsby-transformer-remark plugin prior to versions 5.25.1 and 6.3.2 passes input through to the 'gray-matter' npm package, which is vulnerable to JavaScript injection

QUICK INFO

CVE Dictionary Entry:
CVE-2023-22491
NVD Published Date:
 01/13/2023
NVD Last Modified:
 11/06/2023
Source:

Bước 3: Và có POC liên quan tới lỗi này

Unsanitized JavaScript code injection possible in gatsby-transformer-remark

High mlgualtieri published GHSA-7ch4-rr99-cqcw on Jan 11, 2023

Package gatsby-transformer-remark (npm)

Affected versions <=5.25.0; <=6.3.1

Patched versions 5.25.1; 6.3.2

Severity **High** 8.1 / 10

Description

Impact

The gatsby-transformer-remark plugin prior to versions 5.25.1 and 6.3.2 passes input through to the 'gray-matter' npm package, which is vulnerable to JavaScript injection in its default configuration, unless input is sanitized. The vulnerability is present in gatsby-transformer-remark when passing input in data mode (querying MarkdownRemark nodes via GraphQL). Injected JavaScript executes in the context of the build server.

To exploit this vulnerability untrusted/unsanitized input would need to be sourced by or added into a file processed by gatsby-transformer-remark. The following payload demonstrates a vulnerable configuration:

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

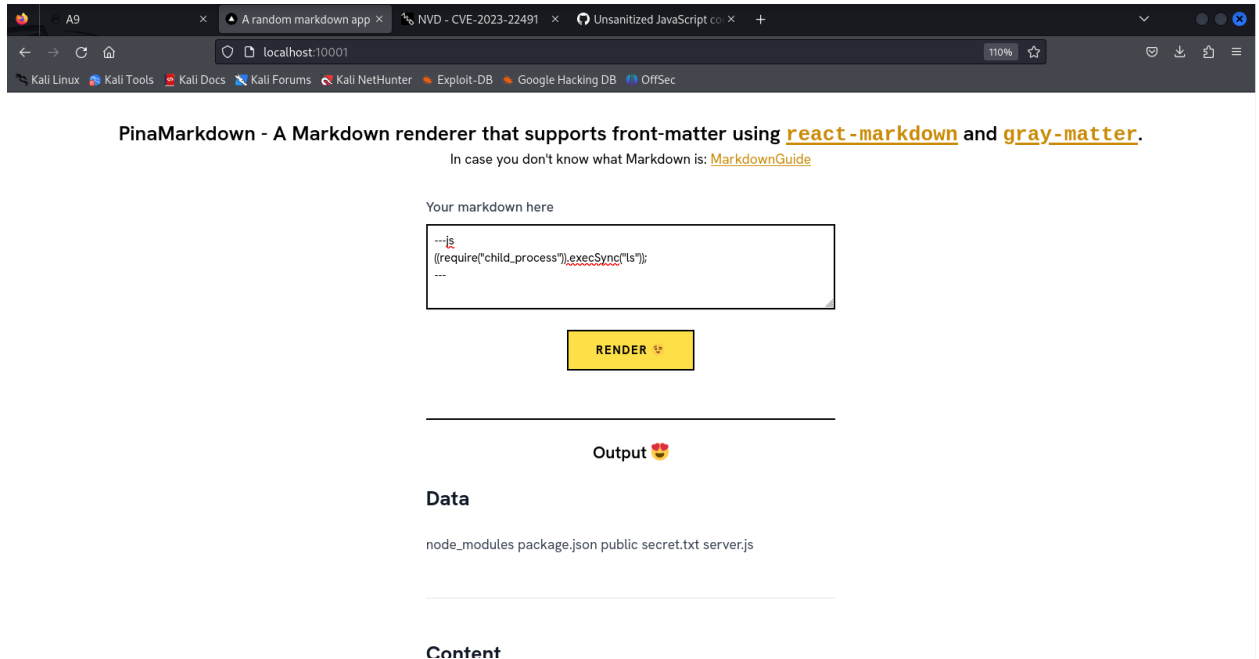
CVSS base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	Low
User interaction	None
Scope	Unchanged
Confidentiality	High
Integrity	High
Availability	None

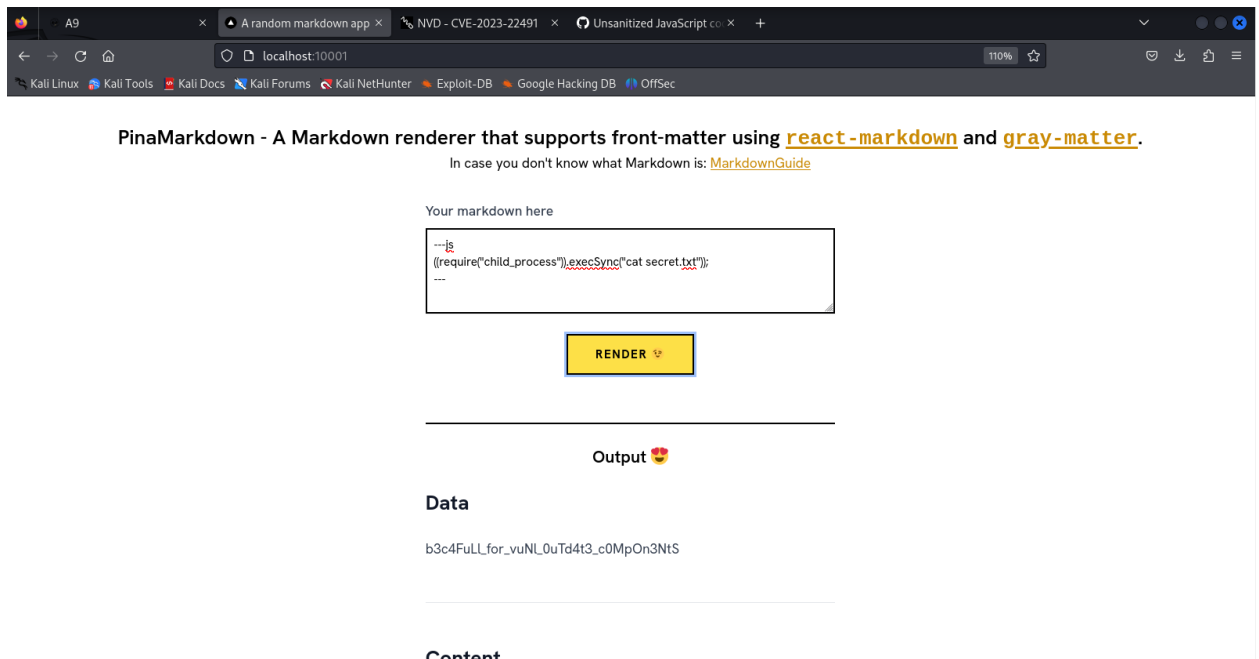
CVSS 3.1 AV:N/AC:L/PRL/UI:N/S:U/C:H/I:H/A:N

CVE ID CVE-2023-22491

Bước 4: Vận dụng POC, ta sửa lại để thực thi lệnh "ls". Kết quả hiển thị cho thấy có 1 file secret.txt



Bước 5: Sửa lại input lần nữa để đọc file secret.txt. Ta được kết quả như sau:



Mức độ ảnh hưởng của lỗ hổng: Cao. Lỗ hổng này có thể gây ra tác động nghiêm trọng đến bảo mật của hệ thống. Các thành phần phần mềm cũ, không được cập nhật đều đặn hoặc chứa các lỗ hổng bảo mật có thể tạo ra các điểm yếu cho kẻ tấn công tấn công vào hệ thống một cách dễ dàng.

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

b) A07:2021 – Identification and Authentication Failures

Chậm lại và suy nghĩ 1: Dựa vào thông tin recon được, có khai thác được gì không, ngoài ra còn có lỗi nào khác không, có thể đọc mã nguồn ứng dụng để tìm hiểu?

Trả lời:

Khi đọc mã nguồn, ta nhận thấy có 1 biến tên là fail_attempt. Mỗi lần user đăng nhập thất bại biến này sẽ tăng lên 1. Nếu quá 5 lần thì tài khoản sẽ bị khóa

```
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,
```

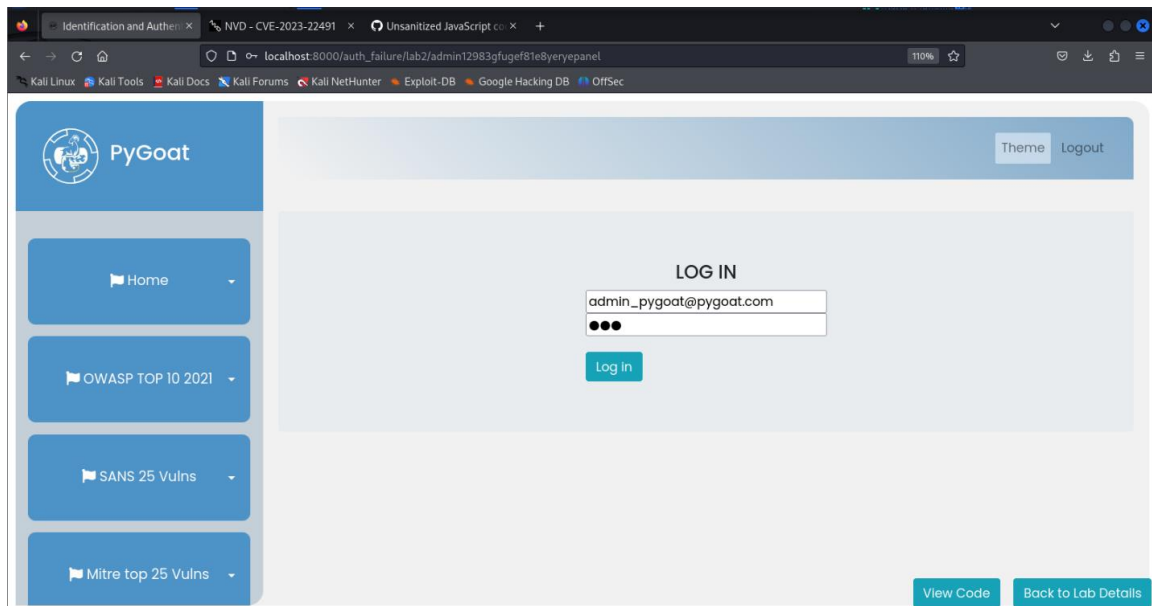
Bài tập 2: Identification and Authentication Failures

Tiêu đề: Identification and Authentication Failures - Lỗ hổng liên quan tới xác minh và cho phép quyền truy cập của người dùng.

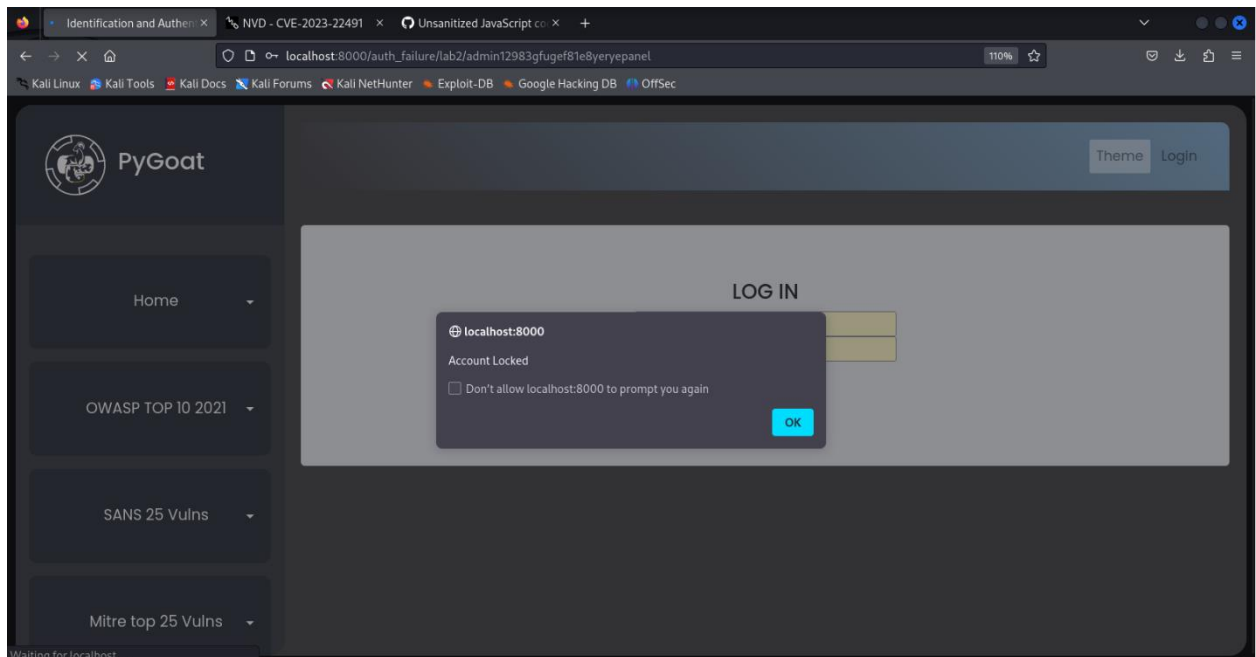
Mô tả lỗ hổng:

- **Tóm tắt:** Trang web không có cơ chế bảo mật người dùng nào ngoài username/password. Điều này dẫn đến tấn công DOS. Trong bài nếu thực hiện brute-force mật khẩu sẽ dẫn đến tài khoản admin bị khóa.
- **Các bước để thực hiện lại và bằng chứng:**

Bước 1: Từ gợi ý của đề bài, ta chỉ biết được username là admin_pygoat@pygoat.com. Tuy nhiên theo phân tích ở đầu bài, ta có thể tạm ngưng quyền của admin bằng cách nhập sai mật khẩu 5 lần. Vậy ta có thể chặn quyền admin mà không cần biết password là gì.



Bước 2: Sau khi nhập mật khẩu sai 5 lần, máy sẽ hiển thị thông báo



Bước 3: Sau khi hiển thị thông báo trên, account sẽ bị khóa trong vòng 1440 phút hay 24h. Vậy là ta đã chặn được quyền admin


```

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,

```

Mức độ ảnh hưởng của lỗ hổng: Cao. Khi các quá trình xác thực và xác nhận danh tính không được triển khai một cách an toàn và hiệu quả, tổ chức có thể mở ra cửa cho các cuộc tấn công từ phía ngoài hoặc từ bên trong hệ thống.

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: Có thêm phương thức đăng nhập qua email, facebook. Lưu lại tên thiết bị đã đăng nhập trước đó. Thông báo cho chủ tài khoản biết có hành vi đăng nhập bất thường và tài khoản đã bị khóa.

c) A08:2021 – Software and Data Integrity Failures

Chậm lại và suy nghĩ 2: Lỗi ở đây là gì, gây nên vấn đề gì đối với chức năng của web thực tế ảnh hưởng đến sự toàn vẹn của phần mềm?

Trả lời:

- Lỗi ở đây là việc không kiểm tra và xử lý đầu vào của người dùng một cách an toàn (lỗi kiểm soát đầu vào). Khi người dùng có thể nhập bất kỳ dữ liệu nào vào ô input mà không có bất kỳ điều kiện kiểm tra nào, điều này tạo ra một lỗ hổng bảo mật lớn, gọi là lỗ hổng chèn mã (code injection).

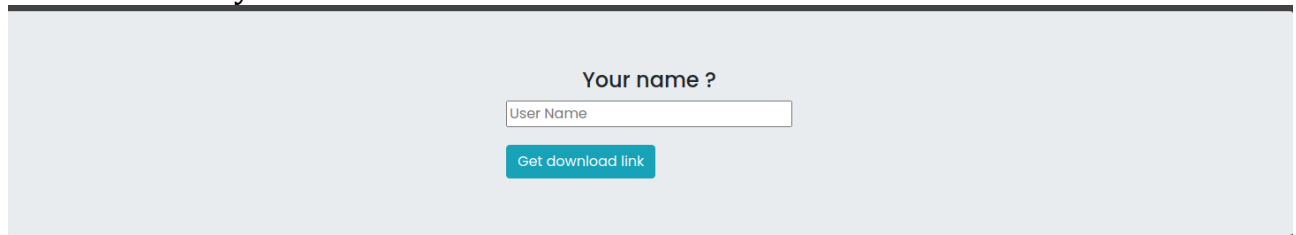
Bài tập 3: Báo cáo lỗ hổng đang được thực hành. Sử dụng format theo mẫu.

Tiêu đề: Software and Data Integrity Failures - Lỗ hổng liên quan đến việc thiếu điều kiện để đảm bảo tính toàn vẹn của phần mềm và dữ liệu.

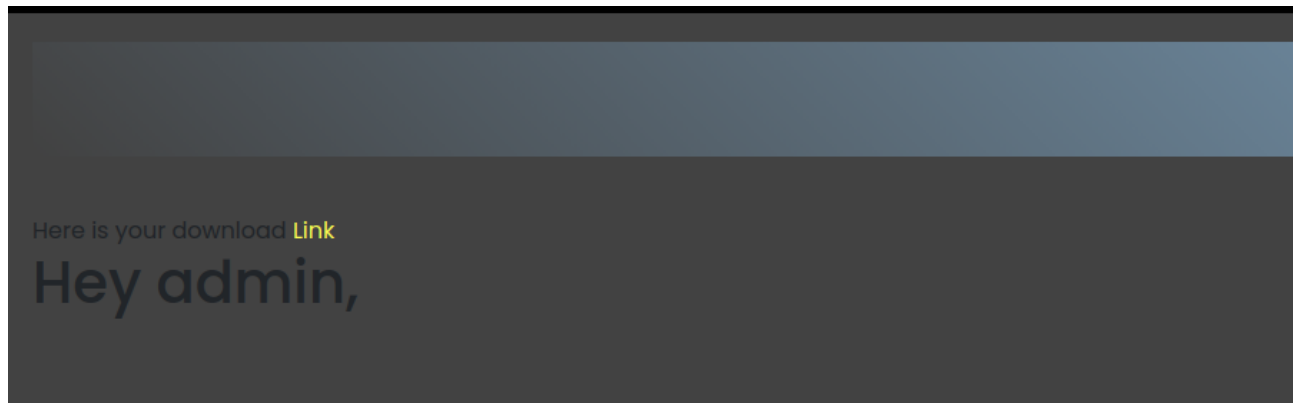
Mô tả lỗ hổng:

- **Tóm tắt:** Lỗ hổng này xuất phát từ việc thiếu điều kiện để đảm bảo tính toàn vẹn của phần mềm và dữ liệu, có thể dẫn đến việc sửa đổi, xâm nhập hoặc phá hủy dữ liệu quan trọng hoặc mã nguồn.
- **Các bước để thực hiện lại và bằng chứng:**

Bước 1: Đăng nhập bằng tên người dùng “admin” và bấm vào “Link” để tải file “real.txt” về máy.



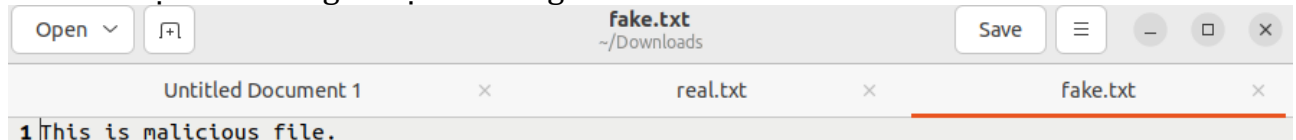
The screenshot shows a web form with the title "Your name ?". Below the title is a text input field labeled "User Name". Below the input field is a blue button labeled "Get download link".



Bước 2: Kiểm tra nội dung file.

```
huynhminhkhue@huynhminhkhue-virtual-machine:~/Downloads$ cat real.txt
This is real file huynhminhkhue@huynhminhkhue-virtual-machine:~/Downloads$
```

Bước 3: Tạo 1 file txt giả mạo để dùng cho các bước sau.

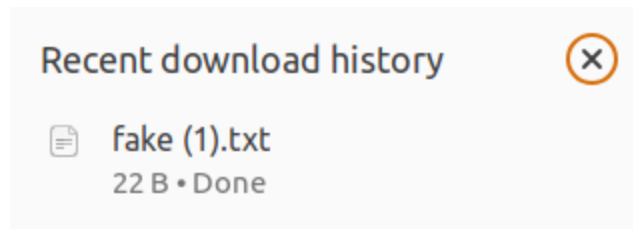


Bước 4: Dùng hàm băm sha256 để băm. Ta thấy kết quả băm khác nhau.

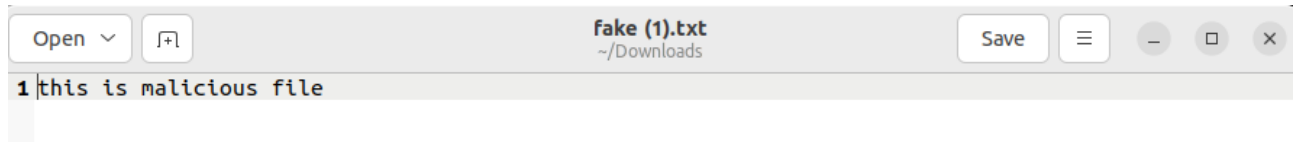
```
huynhminhkhue@huynhminhkhue-virtual-machine:~/Downloads$ sha256sum real.txt
773ff9dfab2f6568bcce2b8f3a8db4e5c6f6962b187e65e5b0896c7cf2cfa777 real.txt
huynhminhkhue@huynhminhkhue-virtual-machine:~/Downloads$ sha256sum fake.txt
b7ec101e0d994f3f09ad26bc1c6b38848fcfc8a2da9a9b4a9e57542d194ee5a5 fake.txt
huynhminhkhue@huynhminhkhue-virtual-machine:~/Downloads$
```

Bước 5: Nhập 1 đoạn mã script vào ô user name. Khi bấm nút download, ta thấy 1 file giả mạo được tải về. Cả tên và nội dung file đều khác file ban đầu (real.txt).

```
<script>document.getElementById("download_link").href =
"/static/fake.txt";</script>
```



File fake.txt được tải về



Nội dung file fake.txt

Mức độ ảnh hưởng của lỗ hổng: Lỗ hổng này có thể gây ra hậu quả nghiêm trọng cho tính toàn vẹn và an ninh của phần mềm và dữ liệu. Kẻ tấn công có thể thay đổi hoặc phá hủy dữ liệu, can thiệp vào hoạt động của phần mềm, hoặc thậm chí đưa ra thông tin sai lệch để đánh lừa người dùng hoặc hệ thống.

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: Để khắc phục lỗ hổng này, cần thiết lập các biện pháp kiểm soát cứng rắn để đảm bảo tính toàn vẹn của phần mềm và dữ liệu, bao gồm việc sử dụng chữ ký số, mã hóa dữ liệu, kiểm tra tính toàn vẹn của dữ liệu, và thiết lập các quy trình kiểm tra mã nguồn an toàn. Đồng thời, việc duy trì và cập nhật định kỳ các bản vá bảo mật cũng là rất quan trọng.

d) A09:2021 – Security Logging and Monitoring Failures

Chậm lại và suy nghĩ 3: Bài thực hành ghi log ở đâu, thông tin nhạy cảm có thể được tiết lộ từ vị trí nào của log?

Trả lời:

Bài thực hành ghi log ở đường dẫn localhost:8000/debug, trong đây chứa các thông tin, yêu cầu, trạng thái,... của trang web. Thông tin nhạy cảm có thể có trong các thông báo, ví dụ “INFO GET username=...&password=....”

Bài tập 4: Báo cáo lỗ hổng đang được thực hành. Sử dụng format theo mẫu.

Tiêu đề: Security Logging and Monitoring Failures - Lỗ hổng liên quan tới ghi nhật ký và giám sát bảo mật.

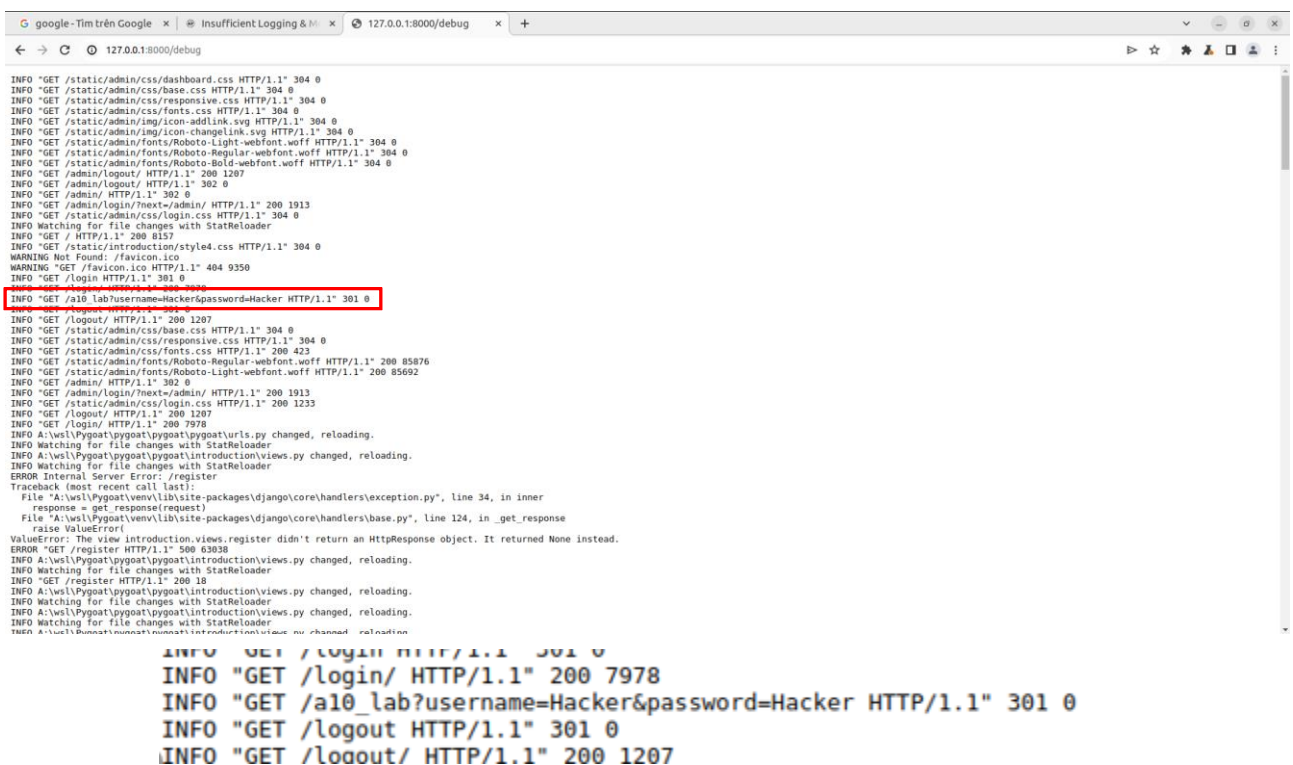
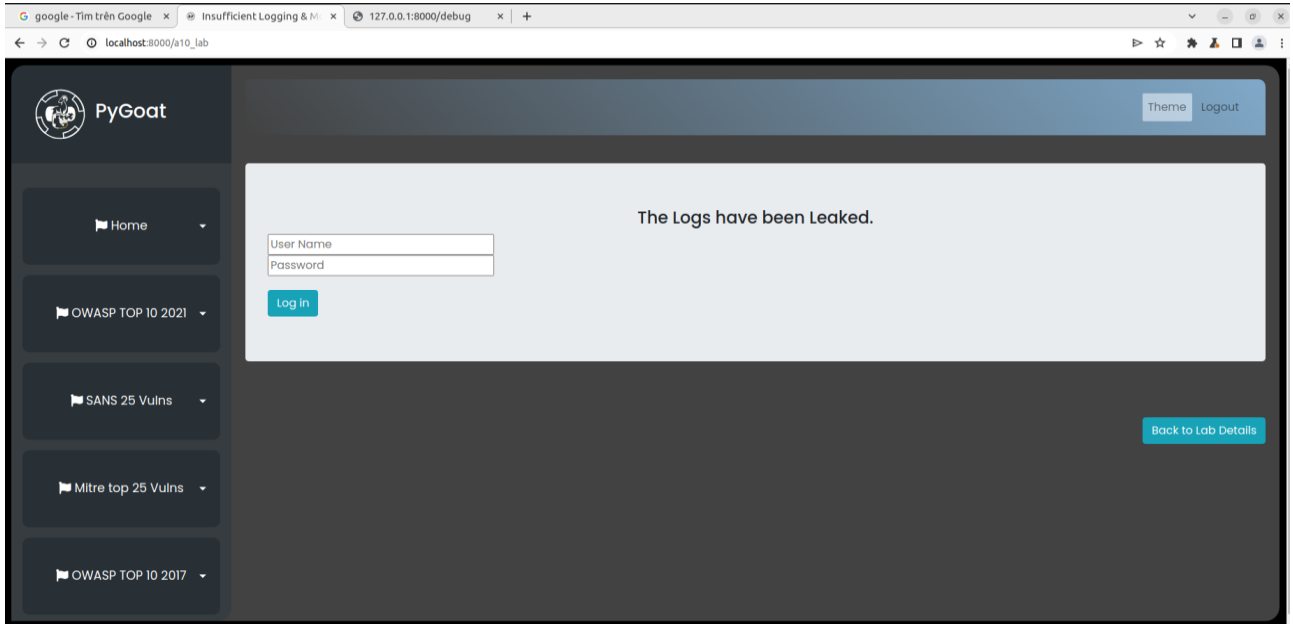
Mô tả lỗ hổng:

- **Tóm tắt:** Lỗ hổng này là do vị trí ghi log ở nơi ai cũng có cũng có thể truy cập được. Trong bài này, tên người dùng, mật khẩu được hiển thị trong các thông báo

của nhật ký, chỉ cần truy cập vào đường dẫn localhost:8000/debug là có thể tìm thấy.

- Các bước để thực hiện lại và bằng chứng:**

Bước 1: Vào đường link localhost:8000/debug. Ta sẽ tìm thấy tên và mật khẩu người dùng.



Bước 2: Có thể dùng tên và mật khẩu đó để đăng nhập.

The Logs have been Leaked.

Hacker

.....

Log in

Mức độ ảnh hưởng của lỗ hổng: Lỗ hổng này có thể gây ra tác động nghiêm trọng đến bảo mật hệ thống và dữ liệu của ứng dụng web. Nếu không có sự giám sát và ghi nhật ký hiệu quả, tổ chức sẽ mất khả năng phát hiện và ngăn chặn các cuộc tấn công mạng.

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: Tổ chức cần triển khai các biện pháp giám sát và ghi nhật ký hiệu quả, bao gồm việc đảm bảo rằng các hành động đáng ngờ hoặc không phù hợp đều được ghi lại và phản ứng kịp thời khi phát hiện có sự vi phạm.

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

Chậm lại và suy nghĩ 4: Vị trí lỗ hổng ở đâu, khai thác lỗi này như thế nào?

Trả lời:

- Lỗ hổng này nằm ở trong file code của trang web. Do việc xử lý điều kiện cho các button không tốt nên để khai thác lỗ hổng này, attacker chỉ cần thay đổi giá trị của các thuộc tính và bấm nút, trang web sẽ trả về thông tin mà attacker mong muốn.

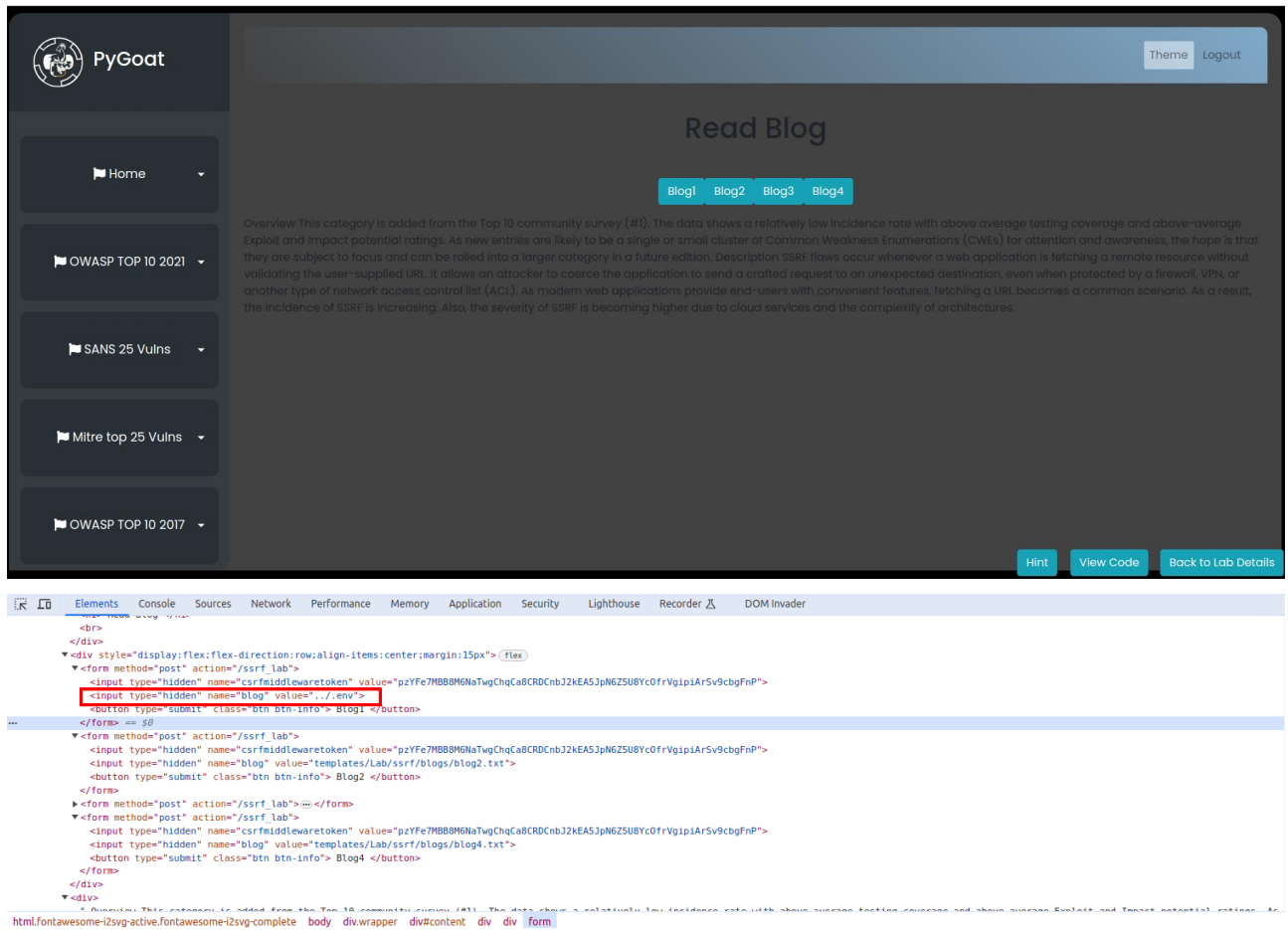
Bài tập 5: Báo cáo lỗ hổng đang được thực hành. Sử dụng format theo mẫu.

Tiêu đề: Server-Side Request Forgery (SSRF) - Lỗ hổng yêu cầu giả mạo phía máy chủ.

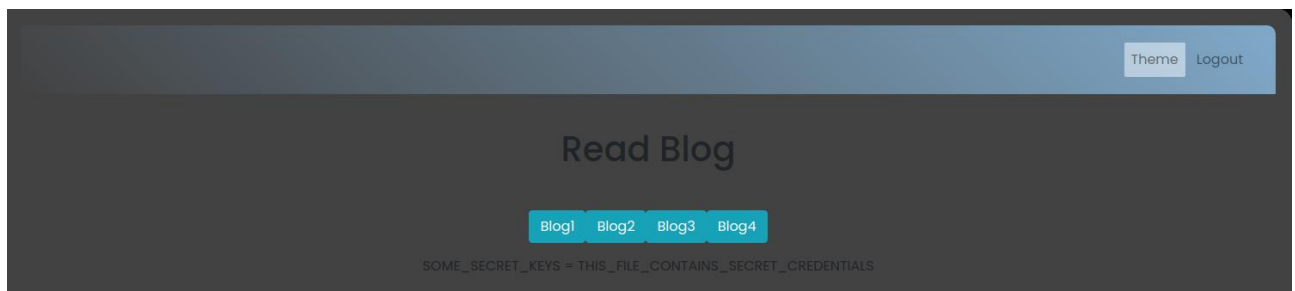
Mô tả lỗ hổng:

- **Tóm tắt:** một lỗ hổng bảo mật trong ứng dụng web, cho phép kẻ tấn công tạo ra các yêu cầu từ máy chủ đích mà không cần sự chấp thuận hoặc kiểm soát từ phía máy chủ.
- **Các bước để thực hiện lại và bằng chứng:**

Bước 1: Nháy chuột phải chọn “Inspect”, cửa sổ code của trang web sẽ hiện ra, cho phép chỉnh sửa code. Tìm và chỉnh sửa button “Blog1”. Ở input có thuộc tính “name” giá trị bằng “blog”, sửa giá trị của thuộc tính “value” thành “../.env”.



Bước 2: Bấm nút “Blog1”, thông tin sẽ thay đổi. Nội dung hiển thị bây giờ không phải là của “Blog1” mà là nội dung file .env.



Mức độ ảnh hưởng của lỗ hổng: SSRF có thể gây ra tác động nghiêm trọng đến bảo mật hệ thống, cho phép kẻ tấn công truy cập vào các tài nguyên nội bộ hoặc không được công khai trên mạng nội bộ, thậm chí có thể tấn công các hệ thống trong mạng nội bộ.

Tác động bảo mật nào mà kẻ tấn công có thể đạt được: Kẻ tấn công có thể sử dụng SSRF để đọc hoặc thay đổi dữ liệu trên máy chủ nội bộ, tấn công các hệ thống nội bộ, hoặc thậm chí khai thác các lỗ hổng khác từ xa.

Khuyến cáo khắc phục: Nên giới hạn quyền truy cập mạng của ứng dụng web, cập nhật các cơ sở dữ liệu địa chỉ IP không an toàn, và áp dụng các biện pháp kiểm soát truy cập.

như giới hạn URL được phép gửi yêu cầu hoặc sử dụng bộ lọc firewall để ngăn chặn các yêu cầu không mong muốn.