

Bộ Giáo Dục Và Đào Tạo

Trường Đại Học Ngoại Ngữ - Tin Học Thành Phố Hồ Chí Minh

**Khoa Công Nghệ Thông Tin**



**ĐỒ ÁN CUỐI KÌ**

**ĐỀ TÀI: WEB PENETRATION TESTING**

**GIẢNG VIÊN HƯỚNG DẪN:** ThS. PHẠM ĐÌNH THẮNG

**SINH VIÊN THỰC HIỆN:**

NGÔ THẾ ĐỨC – 22DH114504

*Tp. Hồ Chí Minh, ngày 1 tháng 4 năm 2025*

## LỜI CẢM ƠN

Tôi xin gửi lời cảm ơn chân thành và sâu sắc đến ThS. Phạm Đình Thắng, người đã tận tình hướng dẫn, hỗ trợ và truyền đạt những kiến thức quý báu trong suốt quá trình tôi thực hiện đồ án này. Nhờ có sự chỉ dạy tận tâm của thầy, tôi không chỉ hiểu rõ hơn về các nguyên tắc lý thuyết mà còn có cơ hội vận dụng chúng vào thực tiễn, rèn luyện tư duy logic, kỹ năng phân tích và giải quyết vấn đề một cách khoa học. Những góp ý và định hướng từ thầy đã giúp tôi hoàn thiện đồ án với chất lượng tốt nhất và tích lũy thêm nhiều kinh nghiệm quý giá trong quá trình nghiên cứu và phát triển.

Tôi cũng xin gửi lời cảm ơn đến quý thầy cô trong khoa, những người đã giảng dạy và trang bị cho tôi nền tảng kiến thức vững chắc trong suốt thời gian học tập. Những bài giảng, những chia sẻ và định hướng của thầy cô không chỉ giúp tôi mở rộng hiểu biết về lĩnh vực mình theo đuổi mà còn tạo động lực để tôi tiếp tục học hỏi và nâng cao năng lực của bản thân.

Việc thực hiện đồ án một mình là một trải nghiệm đầy thử thách nhưng cũng rất ý nghĩa, giúp tôi nâng cao khả năng tự học, tự nghiên cứu và giải quyết vấn đề một cách độc lập. Quá trình này không chỉ giúp tôi củng cố kiến thức chuyên môn mà còn rèn luyện tinh thần trách nhiệm, sự kiên trì trước những khó khăn. Những kinh nghiệm quý báu có được từ quá trình thực hiện đồ án chắc chắn sẽ là hành trang hữu ích, hỗ trợ tôi trong công việc sau này, giúp tôi có thể thích nghi và phát triển tốt hơn trong môi trường làm việc thực tế.

Tôi xin chân thành cảm ơn!

# MỤC LỤC

LỜI CẢM ƠN .....	1
MỤC LỤC .....	2
DANH MỤC ẢNH .....	3
DANH MỤC BẢNG.....	3
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT .....	4
1.1.    Kiểm thử bảo mật (Security Testing).....	4
1.1.1.    Bảo mật (Security) là gì? .....	4
1.1.2.    Kiểm thử bảo mật (Security Testing) là gì ? .....	4
1.1.3.    Quy trình của kiểm thử bảo mật .....	4
1.2.    Khái niệm về Penetration Testing .....	5
1.2.1.    Penetration Testing là gì ? .....	5
1.2.2.    Khái niệm bảo mật cơ bản ?.....	5
1.2.3.    Các phương pháp kiểm thử trong Pentest .....	6
1.2.4.    Tiêu chuẩn thực hiện Pentest .....	7
CHƯƠNG 2: QUY TRÌNH KIỂM THỬ .....	8
2.1.    Các mô hình Pentest.....	8
2.1.1.    OSSTMM – Open Source Security Testing Methodology Manual .....	8
2.1.2.    OWASP – Open Web Application Security Protocol.....	9
2.1.3.    ISSAF.....	10
2.2.    Xây dựng mô hình .....	11
CHƯƠNG 3: THỰC NGHIỆM KIỂM THỬ TRÊN WEB .....	13
3.1. Information Gathering (Thu thập thông tin).....	13
3.2. Kiểm thử A01: Broken Access Control .....	17
3.3. Kiểm thử A03: Injection .....	21
3.4. Kiểm thử A04: Insecure Design .....	23
3.5. Kiểm thử A05: Security Misconfiguration .....	24
3.6. Kiểm thử A06: Vulnerable Components .....	26
CHƯƠNG 4: KẾT LUẬN .....	27
TÀI LIỆU THAM KHẢO .....	28

## DANH MỤC ẢNH

Hình 1. Thu thập thông tin về cấu hình .....	13
Hình 2. Thông tin bên trong .....	13
Hình 3. Sử dụng whois lấy thông tin .....	14
Hình 4. HTTP Header .....	14
Hình 5. Nmap .....	15
Hình 6. Lỗ hổng A04 - High .....	15
Hình 7. Lỗ hổng A01 – Medium .....	16
Hình 8. Lỗ hổng A05 – Medium .....	16
Hình 9. Lỗ hổng A06 – Medium .....	17
Hình 10. Giao diện đăng nhập .....	17
Hình 11. Giao diện điện thoại .....	18
Hình 12. Đăng nhập tài khoản ngẫu nhiên .....	19
Hình 13. Đăng nhập đúng tài khoản .....	19
Hình 14. Brute force .....	20
Hình 15: Thu thập các số điện thoại .....	20
Hình 21. Request của đường dẫn .....	21
Hình 22. Thêm đoạn mã XSS .....	22
Hình 23. Định dạng lại .....	22

## DANH MỤC BẢNG

## CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

### 1.1. Kiểm thử bảo mật (Security Testing)

#### 1.1.1. Bảo mật (Security) là gì?

Bảo mật (Security) là các biện pháp thiết lập để bảo mật một ứng dụng chống lại các hành động không lường trước được, các hành động đó sẽ ảnh hưởng hoặc phá hủy ứng dụng. Hành động không lường trước có thể là cố ý hoặc vô ý

#### 1.1.2. Kiểm thử bảo mật (Security Testing) là gì ?

Kiểm thử bảo mật (Security Testing) là việc tìm kiếm tất cả các lỗ hổng có thể và điểm yếu trong hệ thống nhằm đảm bảo rằng các hệ thống và ứng dụng trong một tổ chức không có bất kì sơ hở nào có thể gây ra tổn thất về an toàn bảo mật mà dẫn đến rò rỉ thông tin của tổ chức

#### 1.1.3. Quy trình của kiểm thử bảo mật

Có 4 bước chính trong quá trình kiểm thử bảo mật, quá trình này được lặp đi lặp lại nhiều lần



## 1.2. Khái niệm về Penetration Testing

### 1.2.1. Penetration Testing là gì ?

Pentest (Penetration Testing) là hình thức kiểm tra hệ thống công nghệ thông tin của người dùng có thể bị tấn công hay không, bằng cách giả lập các vụ tấn công thử nghiệm. Có thể hiểu một cách đơn giản, Pentest chính là đánh giá độ an toàn bằng cách tấn công vào hệ thống, là quá trình xem xét lại các dịch vụ và hệ thống để tìm ra các vấn đề an ninh tiềm tàng hoặc dò tìm các dấu vết khi hệ thống bị tổn thương. Người thực hiện một thử nghiệm xâm nhập được gọi là kiểm tra xâm nhập hoặc Pentester

### 1.2.2. Khái niệm bảo mật cơ bản ?

- Lỗ hổng (Vulnerabilities)

Vulnerabilities là lỗ hổng bảo mật trong một phần của phần mềm, phần cứng hoặc hệ điều hành, cung cấp một góc tiềm năng để tấn công hệ thống. Một lỗ hổng có thể đơn giản như mật khẩu yếu hoặc phức tạp như lỗi tràn bộ đệm hoặc các lỗ hổng SQL injection.

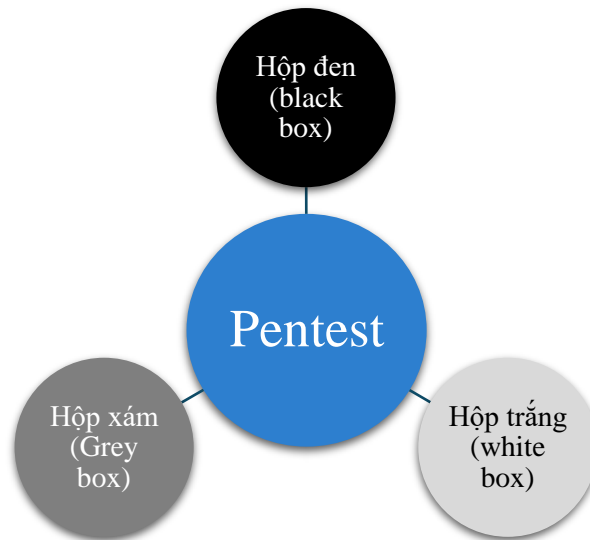
- Khai thác (Exploits)

Để tận dụng lợi thế của một lỗ hổng, thường cần một sự khai thác, một chương trình máy tính nhỏ và chuyên môn cao mà lý do duy nhất là để tận dụng lợi thế của một lỗ hổng cụ thể và để cung cấp truy cập vào một hệ thống máy tính. Khai thác thường cung cấp một tải trọng (payloads) đến mục tiêu hệ thống và cung cấp cho kẻ tấn công truy cập vào hệ thống.

- Trọng tải (Payload)

Trọng tải (payloads) là các thành phần của phần mềm cho phép kiểm soát một hệ thống máy tính sau khi nó đang được khai thác lỗ hổng ,thường gắn liền với vài giao khai thác (exploits).

### 1.2.3. Các phương pháp kiểm thử trong Pentest



- Hộp đen (Black box)

Tấn công từ ngoài vào (Black-box Pentest): các cuộc tấn công được thực hiện mà không có bất kỳ thông tin nào, pentester sẽ đặt mình vào vị trí của những tin tặc mũ đen và cố gắng bằng mọi cách để thâm nhập vào được mạng nội, ngoại của khách hàng.

Pentester sẽ mô phỏng một cuộc tấn công thực sự vào ứng dụng, quá trình thử nghiệm bao gồm một loạt các lỗ hổng bảo mật ở cấp ứng dụng được xác định bởi OWASP và WASC, nhằm mục tiêu các lỗ hổng bảo mật nguy hiểm tiềm tàng trong ứng dụng của khách hàng. Quá trình thử nghiệm sẽ tiết lộ các lỗ hổng, thiệt hại khai thác tiềm năng và mức độ nghiêm trọng.

- Hộp trắng (White box)

Tấn công từ trong ra (White-box Pentest): là phương pháp kiểm thử bảo mật trong đó khách hàng cung cấp thông tin về hệ thống mạng nội bộ và bên ngoài cho Pentester.

Mục tiêu là đánh giá rủi ro từ những người có quyền truy cập, như nhân viên, khách hàng hoặc đối tác. Những đối tượng này có thể sở hữu thông tin quan trọng về hệ thống, ứng dụng, cơ cấu tổ chức, giúp họ thực hiện các cuộc tấn công có chủ đích. White-box Pentest giúp xác định các lỗ hổng mà kiểm thử Black-box có thể bỏ sót.

- Hộp xám (Gray box)

Kiểm định hộp xám (Gray-box hay Crystal-box): Giả định như tin tặc được cung cấp tài khoản một người dùng thông thường và tiến hành tấn công vào hệ thống như một nhân viên của doanh nghiệp.

#### 1.2.4. Tiêu chuẩn thực hiện Pentest

Đánh giá ứng dụng web – OWASP (Open-source Web Application Security Project): OWASP là một chuẩn mở cho phép tổ chức/ doanh nghiệp tiến hành xây dựng, phát triển, duy trì hoạt động của các ứng dụng trên nền tảng web một cách bảo mật nhất. Quá trình đánh giá dựa trên các tiêu chí đã được cộng đồng bảo mật xác nhận. OWASP bao gồm 10 lỗ hổng được đánh giá một cách chi tiết và cập nhật thường xuyên với thực tế các nguy cơ mà một ứng dụng web thường gặp.

Đánh giá mạng và hệ thống – OSSTMM (Open Source Security Testing Methodology Manual): OSSTMM là một chuẩn mở cung cấp phương pháp kiểm tra bảo mật một hệ thống đang hoạt động của doanh nghiệp. Từ phiên bản 3.0, OSSTMM đưa ra phương pháp kiểm định cho hầu hết các thành tố trong hệ thống như: con người, hạ tầng vật lý, mạng không dây, truyền thông và các mạng sử dụng truyền dữ liệu



## CHƯƠNG 2: QUY TRÌNH KIỂM THỬ

### 2.1. Các mô hình Pentest

#### 2.1.1. OSSTMM – Open Source Security Testing Methodology Manual

Open Source Security Testing Methodology Manual (OSSTMM) là một chuẩn mở cung cấp một phương pháp kiểm tra an ninh toàn diện cho một hệ thống mạng.

- Mục đích ra đời

Trong những năm qua do môi trường mạng ngày càng phức tạp như việc điều khiển thiết bị từ xa qua mạng, ảo hóa, điện toán đám mây và các loại cơ sở hạ tầng mới, việc bảo đảm an ninh không còn dừng lại ở các thử nghiệm đơn giản trên máy bàn, máy chủ, các thiết bị định tuyến. Do đó, từ phiên bản 3, OSSTMM audit đã bao quát tất cả lĩnh vực bao gồm: yếu tố con người, đường truyền vật lý, thiết bị không dây, thiết bị viễn thông và dữ liệu mạng.

- Nổi bật

OSSTMM phù hợp với nhiều hình thức kiểm tra bảo mật như: đánh giá lỗ hổng, kiểm thử thâm nhập, kiểm thử hộp trắng,....

Phương pháp đánh giá này được thực hiện triệt để và kết quả báo cáo đầy đủ, có định lượng và đáng tin cậy.



- Phổ dụng

OSSTMM sử dụng phương pháp RAV (Risk Assessment Values) do đó chúng ta có thể xác định được mức độ rủi ro. Căn cứ vào hoạt động an ninh, kiểm soát sự rủi ro và các giới hạn, RAV có thể tính toán được giá trị bảo mật thực tế. Số điểm RAV đưa ra tương đương với trạng thái an ninh hiện tại của đối tượng.

Mặt khác, những báo của của OSSTMM được trình bày dưới dạng STAR nên đôi ngũ quản lý dễ dàng phân tích, đánh giá hệ thống.

- Quy trình kiểm thử của OSSTMM dễ dàng phù hợp với những quy định của các ngành công nghiệp, ngành kinh doanh và pháp luật, chính phủ.

### 2.1.2. OWASP – Open Web Application Security Protocol

Open Web Application Security Protocol (OWASP) đây là dự án được phát triển bởi cộng đồng mở nhằm nâng cao nhận thức bảo mật ứng dụng trong các tổ chức.

- Mục đích ra đời

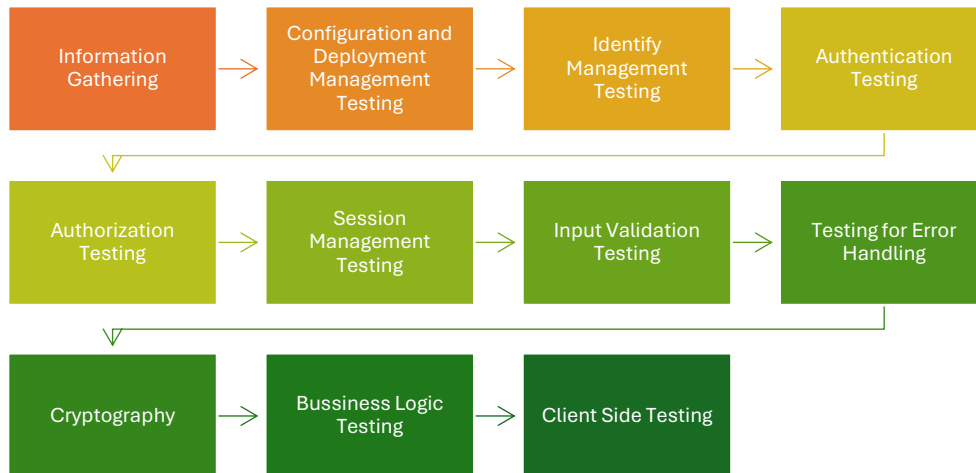
Tổ chức phòng thủ ở các thiết bị mạng không chỉ giúp ngăn chặn mã độc xâm nhập vào mạng bằng cách khai thác thông tin và lỗ hổng, mà còn giúp chủ động ngăn cản những truy cập trái phép và không phù hợp vào hệ thống. Tuy nhiên, điều này không giúp các ứng dụng web tránh khỏi các cuộc tấn công, tin tặc có thể tấn công vào ứng dụng trước khi thực hiện tấn công vào hệ thống. Do vậy, cần có phương pháp kiểm tra, đánh giá các nguy cơ bảo mật cơ bản trên ứng dụng. OWASP được thực hiện với mục tiêu đó.

- Nội bật

Tổ chức Open Web Application Security Project (OWASP) là tổ chức phi lợi nhuận, tổ chức này đã đưa ra chuẩn OWASP phục vụ cho công việc pentest hiệu quả và chi tiết.

- Cung cấp các công cụ và những tiêu chuẩn về An Toàn Thông Tin hàng đầu thế giới.

- Cung cấp các thư viện và tiêu chuẩn trong việc kiểm soát, quản lý an ninh thông tin.
- Cung cấp những tài liệu hàng đầu về bảo mật ứng dụng, pentest, kiểm tra mã nguồn và lập trình một cách an toàn nhất



- **Phổ dụng**

Tiêu chuẩn đánh giá an ninh mạng OWASP cung cấp chi tiết những kỹ thuật đánh giá giúp pentester tiết kiệm thời gian, có cách thức tiến hành kiểm tra hiệu quả.

OWASP hỗ trợ các công cụ kiểm thử cho webapp tự động như: WebScarab, Wapiti, JbroFuzz, SQLiX

- Ngoài tiêu chuẩn pentest OWASP còn có tiêu chuẩn đánh giá mạng và hệ thống OSSTMM, NIST...

### 2.1.3. ISSAF

- **Mục đích ra đời**

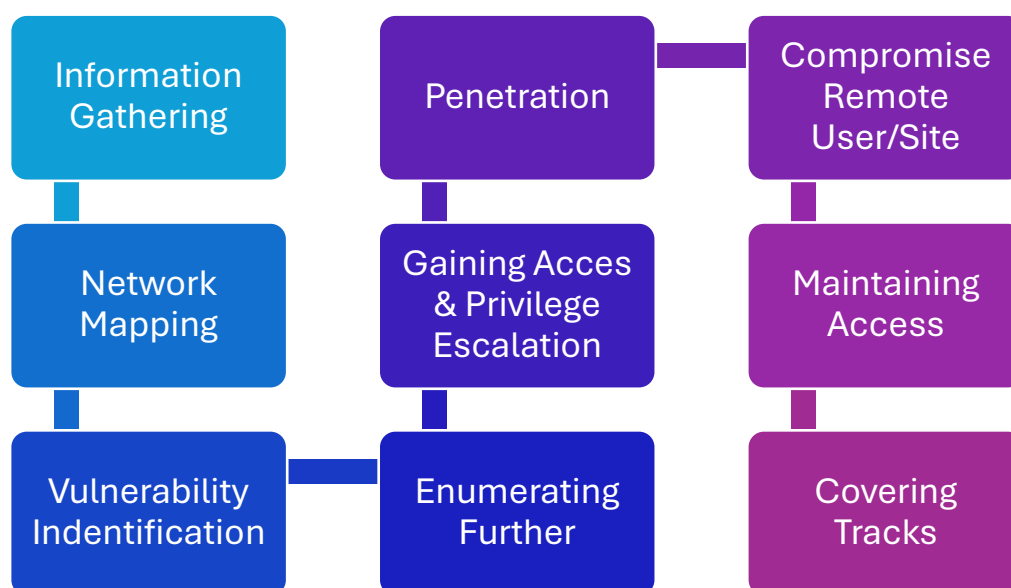
ISSAF là một khung được cung cấp bởi Nhóm bảo mật hệ thống thông tin mở (OISSG), một tổ chức phi lợi nhuận có trụ sở tại London.

Khi pentesting phát triển, các công ty phải hiểu những dịch vụ nào cần thiết, những gì họ cần, những tài sản nào dễ bị tổn thương hoặc những mối đe dọa nào họ có thể gặp phải. Tất cả những điểm đó phải được cả người quản lý và khách hàng của pentesters hiểu rõ, để giải quyết nhu cầu và nhu cầu của khách hàng

- Nổi bật

Đánh giá trước mô tả việc lập kế hoạch và chuẩn bị thử nghiệm. Khía cạnh pháp lý được giải thích rõ ràng và các dòng chính của sự sắp xếp được đưa ra

Việc đánh giá bao gồm các thử nghiệm khác nhau sẽ được tiến hành, tuy nhiên, không có kỹ thuật, tuy nhiên, phương pháp, mục đích và mục tiêu của mỗi và mọi tình huống được đưa ra



- Phổ dụng

ISSAF nổi tiếng là cung cấp một vị trí có giá trị cao về việc đánh giá các kiểm soát bảo mật hiện tại (Shrestha, 2012) và để kết nối các nhiệm vụ giữa chúng. Đối với một pentester mới bắt đầu, nó cung cấp một goldmine, tuy nhiên các pentest được đào tạo sẽ muốn tự mình chuyển sang OSSTMM, một phương pháp khác, cung cấp cho lessexamples và nhiều danh sách đạn hơn, để giữ cho nội dung ở mức nhỏ hơn

## 2.2. Xây dựng mô hình

Mô hình kiểm thử được dựa trên Framework OWASP Top 10

*Bảng 1. Mô hình kiểm thử*

Information Gathering (thu thập thông tin)	Thu thập thật nhiều thông tin	Tools
A01: Broken Access Control	Truy cập trái phép API, thay đổi session ID	Burp Suite, Postman













A02: Cryptographic Failures	Mật khẩu lưu dưới dạng MD5, SHA1	Hashcat, John the Ripper
A03: Injection (SQLi, XSS, NoSQL)	Chèn SQL, thực thi XSS trên form input	SQLmap, XSS Hunter
A04: Insecure Design	Kiểm tra logic bảo mật của ứng dụng	Kiểm tra code thủ công
A05: Security Misconfiguration	Lỗi headers HTTP, debug mode bật	Nikto, Nmap, OpenVAS
A06: Vulnerable Components	Dùng thư viện lỗi thời như Log4j, Apache Struts	OWASP Dependency-Check
A07: Identification & Authentication Failures	Tấn công Brute Force, JWT token	Hydra, Burp Suite
A08: Software Integrity Failures	Kiểm tra supply chain attack	EvilGrade
A09: Logging & Monitoring Failures	Kiểm tra thiếu log, không có alert khi tấn công	Splunk, Graylog
A10: Server-Side Request Forgery (SSRF)	Dùng server gửi request đến nội bộ	SSRFmap, Burp Collaborator

## CHƯƠNG 3: THỰC NGHIỆM KIỂM THỬ TRÊN WEB

### 3.1. Information Gathering (Thu thập thông tin)





Trang web thực hiện PenTesting: <https://vetc.com.vn/>

Lấy thông tin về cấu hình

<b>CMS</b>	<b>Documentation</b>
 Mousewheel JS	 Elevio
<b>Widgets</b>	<b>Analytics</b>
 OWL Carousel	 Google Analytics UA
<b>Web Framework</b>	<b>Web Server</b>
 Bootstrap	<b>IIS IIS 8.5</b>
<b>Programming Language</b>	<b>Operating System</b>
 ASP.NET 4.0.30319	 Windows Server
<b>CDN</b>	<b>Javascript Frameworks</b>
 CloudFlare	 jQuery 1.6.2
 CDN JS	 jQuery Parallax JS 

Hình 1. Thu thập thông tin về cấu hình

Lấy thông tin bên trong

 <b>Hosting Provider:</b> ODS Joint Stock Company	 <b>IP Address:</b> 103.15.50.234
<b>Nameservers:</b> alla.ns.cloudflare.com christian.ns.cloudflare.com	 <b>Owner Details:</b> <a href="#">Whois Record</a> 
<b>Autonomous System Number:</b> 45538	<b>Autonomous System Organization:</b> ODS Joint Stock Company
<b>Organization:</b> ODS Joint Stock Company	<b>Continent:</b> Asia
<b>Country:</b> Vietnam	<b>Registered Country:</b> Vietnam
<b>Location:</b> Asia/Bangkok	

Hình 2. Thông tin bên trong

```

L$ whois 103.15.50.234
% [whois.apnic.net]
% Whois data copyright terms    http://www.apnic.net/db/copyright.html
% Information related to '103.15.48.0 - 103.15.51.255'
% Abuse contact for '103.15.48.0 - 103.15.51.255' is 'hm-changed@vnnic.vn'

inetnum:        103.15.48.0 - 103.15.51.255
netname:        MATBAO-VN
descr:          Mat Bao Corp
descr:          3th Floor, Anna Building, Quang Trung Software Park, Tan Chanh Hiep ward, 12 district, Ho Chi Minh City
admin-c:        NQV2-AP
tech-c:         PKN4-AP
remarks:        send spam and abuse report to info@matbao.com
country:        VN
mnt-by:         MAINT-VN-VNNIC
mnt-lower:      MAINT-VN-VNNIC
mnt-irt:        IRT-VNNIC-AP
status:         ALLOCATED PORTABLE
last-modified:  2017-11-06T08:55:49Z
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:        NTTTT-AP
tech-c:         NTTTT-AP
auth:          # Filtered
mnt-by:         MAINT-VN-VNNIC
last-modified:  2017-11-08T09:40:06Z
source:         APNIC

person:         Nguyen Quoc Vinh
nic-hdl:        NQV2-AP
e-mail:         vinhng@matbao.com
address:        MATBAO-VN
address:        Mat Bao Corp
address:        3th Floor, Anna Building, Quang Trung Software Park, Tan Chanh Hiep ward, 12 district, Ho Chi Minh City
phone:          +84-28-38681999
fax-no:         +84-28-38689983

```

Hình 3. Sử dụng whois lấy thông tin

## Lấy thông tin HTTP Headers

```

HTTP/2 200 OK
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 90778
Content-Type: text/html; charset=UTF-8
Date: Tue, 01 Apr 2025 01:36:49 GMT
Server: Apache

<!DOCTYPE html>

```

Hình 4. HTTP Header

## Kiểm tra các dịch vụ đang chạy

```
(thebduck@thebduck)-[~]
$ nmap -p 1-1000 -T4 -A -v vetc.com.vn
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-03-31 21:42 EDT
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 21:42
Completed NSE at 21:42, 0.00s elapsed
Initiating NSE at 21:42
Completed NSE at 21:42, 0.00s elapsed
Initiating NSE at 21:42
Completed NSE at 21:42, 0.00s elapsed
Initiating Ping Scan at 21:42
Scanning vetc.com.vn (103.15.50.234) [4 ports]
Completed Ping Scan at 21:42, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 21:42
Completed Parallel DNS resolution of 1 host. at 21:42, 0.28s elapsed
Initiating SYN Stealth Scan at 21:42
Scanning vetc.com.vn (103.15.50.234) [1000 ports]
Discovered open port 21/tcp on 103.15.50.234
Discovered open port 22/tcp on 103.15.50.234
Discovered open port 80/tcp on 103.15.50.234
Discovered open port 443/tcp on 103.15.50.234
Discovered open port 887/tcp on 103.15.50.234
Increasing send delay for 103.15.50.234 from 0 to 5 due to 111 out of 276 dropped probes since last increase.
Discovered open port 888/tcp on 103.15.50.234
Increasing send delay for 103.15.50.234 from 5 to 10 due to 227 out of 566 dropped probes since last increase.
Warning: 103.15.50.234 giving up on port because retransmission cap hit (6).
Completed SYN Stealth Scan at 21:43, 61.44s elapsed (1000 total ports)
Initiating Service scan at 21:43
Scanning 6 services on vetc.com.vn (103.15.50.234)
Completed Service scan at 21:43, 12.07s elapsed (6 services on 1 host)
Initiating OS detection (try #1) against vetc.com.vn (103.15.50.234)
Retrying OS detection (try #2) against vetc.com.vn (103.15.50.234)
WARNING: OS didn't match until try #2
Initiating Traceroute at 21:43
Completed Traceroute at 21:43, 0.02s elapsed
Initiating Parallel DNS resolution of 1 host. at 21:43
Completed Parallel DNS resolution of 1 host. at 21:43, 0.00s elapsed
NSE: Script scanning 103.15.50.234.
Initiating NSE at 21:43
Completed NSE at 21:43, 5.07s elapsed
Initiating NSE at 21:43
Completed NSE at 21:43, 1.24s elapsed
Initiating NSE at 21:43
```

Hình 5. Nmap

Quét lỗ hổng:

Edit Alert

URL:https://vetc.com.vn/huong-dan-su-dung-ung-dung-vetc-n3.html

Risk:High

Confidence:High

Parameter:

Attack:

Evidence:4554030296653

CWE ID:359

WASC ID:13

Description:

The response contains Personally Identifiable Information, such as CC number, SSN and similar sensitive data.

Other Info:

Credit Card Type detected: Visa  
Bank Identification Number: 455403  
Brand: VISA

Solution:

Check the response for the potential presence of personally identifiable information (PII), ensure nothing sensitive is leaked by the application.

Reference:

Alert Tags:

Key	Value
OWASP_2021_A04	https://owasp.org/Top10/A04_2021-Insecure_Design/
OWASP_2017_A03	https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure.html
CWE-359	https://cwe.mitre.org/data/definitions/359.html

Cancel

Save

Hình 6. Lỗ hổng A04 - High



**Edit Alert**

URL: <https://vetc.com.vn/lien-he.html>

Risk: Medium

Confidence: Low

Parameter:

Attack:

Evidence: <form method="post" action="index.php?module=contact&view=contact&task=save" name="contact" class="form" enctype="multipart/form-data">

CWE ID: 352

WASC ID: 9

Description:  
No Anti-CSRF tokens were found in a HTML submission form.  
A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast,

Other Info:  
No known Anti-CSRF token [anticsrf, CSRFToken, \_\_RequestVerificationToken, csrfmiddlewaretoken, authenticity\_token, OWASP\_CSRFTOKEN, anoncsrf, csrf\_token, \_csrf, \_csrfSecret, \_csrf\_magic, CSRF, \_token, \_csrf\_token] was found in the following HTML form: [Form 1: "contact\_address" "contact\_email" "contact\_name" "contact\_phone" "contact\_tieude" "itemid" "module" "return" "task" "view"]

Solution:  
Phase: Architecture and Design  
Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.  
For example, use anti-CSRF packages such as the OWASP CSRFGuard.

Reference:  
[https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site\\_Request\\_Forgery\\_Prevention\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html)  
<https://cwe.mitre.org/data/definitions/352.html>

Alert Tags:

Key	Value
OWASP_2021_A01	<a href="https://owasp.org/Top10/A01_2021-Broken_Access_Control/">https://owasp.org/Top10/A01_2021-Broken_Access_Control/</a>
WSTG-v42-SESS-05	<a href="https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/06-Session_Management/05-Session_Token_Management.html">https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/06-Session_Management/05-Session_Token_Management.html</a>
OWASP_2017_A05	<a href="https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control.html">https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control.html</a>

Cancel Save

Hình 7. Lỗ hổng A01 – Medium

**Edit Alert**

Content Security Policy (CSP) Header Not Set

URL: <https://vetc.com.vn/sitemap.xml>

Risk: Medium

Confidence: High

Parameter:

Attack:

Evidence:

CWE ID: 693

WASC ID: 15

Description:  
Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that

Other Info:

Solution:  
Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

Reference:  
[https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing\\_Content\\_Security\\_Policy](https://developer.mozilla.org/en-US/docs/Web/Security/CSP/Introducing_Content_Security_Policy)  
[https://cheatsheetseries.owasp.org/cheatsheets/Content\\_Security\\_Policy\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html)  
<https://www.w3.org/TR/CSP/>

Alert Tags:

Key	Value
CWE-693	<a href="https://cwe.mitre.org/data/definitions/693.html">https://cwe.mitre.org/data/definitions/693.html</a>
OWASP_2021_A05	<a href="https://owasp.org/Top10/A05_2021-Security_Misconfiguration/">https://owasp.org/Top10/A05_2021-Security_Misconfiguration/</a>

Cancel Save

Hình 8. Lỗ hổng A05 – Medium

**Edit Alert**

URL: <https://vetc.com.vn/templates/default/js/bootstrap.min.js?v=7.31>

Risk: Medium

Confidence: Medium

Parameter:

Attack:

Evidence: \* Bootstrap v3.3.1

CWE ID: 1395

WASC ID: 0

Description:  
The identified library appears to be vulnerable.

Other Info:  
The identified library bootstrap, version 3.3.1 is vulnerable.  
CVE-2018-14041  
CVE-2019-8331

Solution:  
Upgrade to the latest version of the affected library.

Reference:  
[https://owasp.org/Top10/A06\\_2021-Vulnerable\\_and\\_Outdated\\_Components/](https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/)

Alert Tags:

Key	Value
CVE-2016-10735	<a href="https://nvd.nist.gov/vuln/detail/CVE-2016-10735">https://nvd.nist.gov/vuln/detail/CVE-2016-10735</a>
OWASP_2021_A06	<a href="https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/">https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components/</a>

Cancel Save

Hình 9. Lỗ hổng A06 – Medium

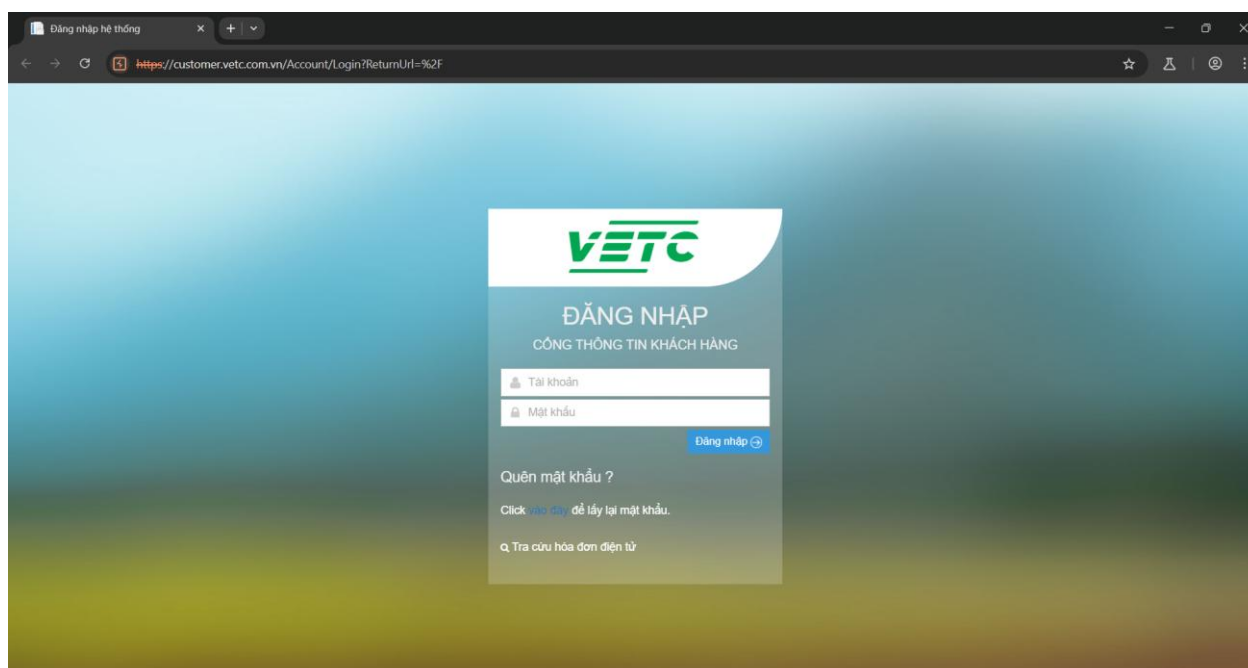
Từ các thông báo lỗ hổng trên có thể thấy còn có khả năng trang Web bị tấn công bằng XSS và đây là lỗ hổng A03: Injection

Từ bước tìm kiếm thông tin ta sẽ bắt đầu kiểm thử các lỗ hổng tìm được

### 3.2. Kiểm thử A01: Broken Access Control

Truy cập vào trang đăng nhập:

<https://customer.vetc.com.vn/Account/Login?ReturnUrl=%2F>



Hình 10. Giao diện đăng nhập

Trang web yêu cầu tạo tài khoản bằng điện thoại

Và ta biết được là :

- Tài khoản: 10 số
- Mật khẩu: 6 số

12:45 2.12 KB/s 13%

< Đăng nhập

Xin chào 0356094640

Mật khẩu

.....

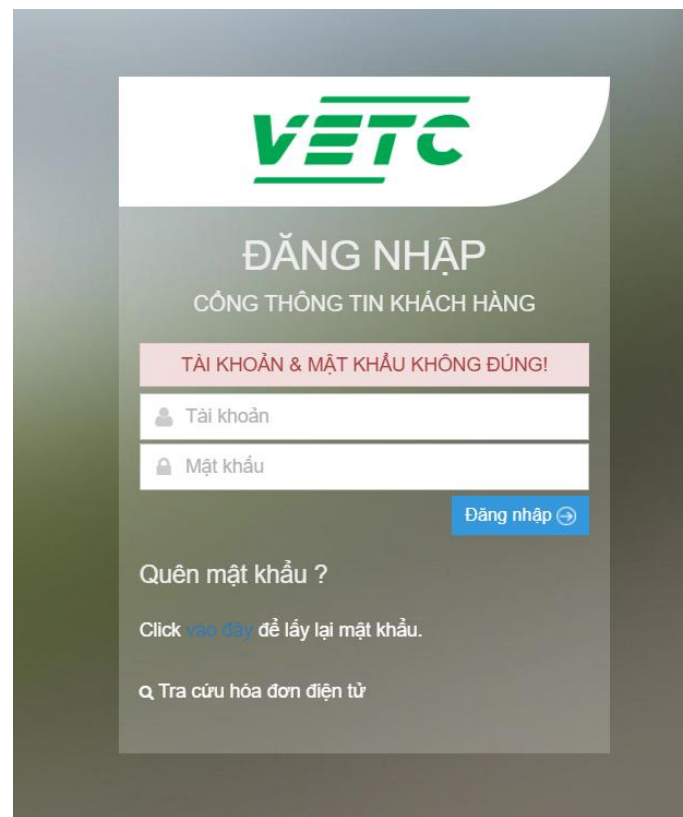
Xác thực bằng vân tay

Đăng nhập

Đổi tài khoản | Quên mật khẩu

Hình 11. Giao diện điện thoại

Đăng nhập với tài khoản bất kì

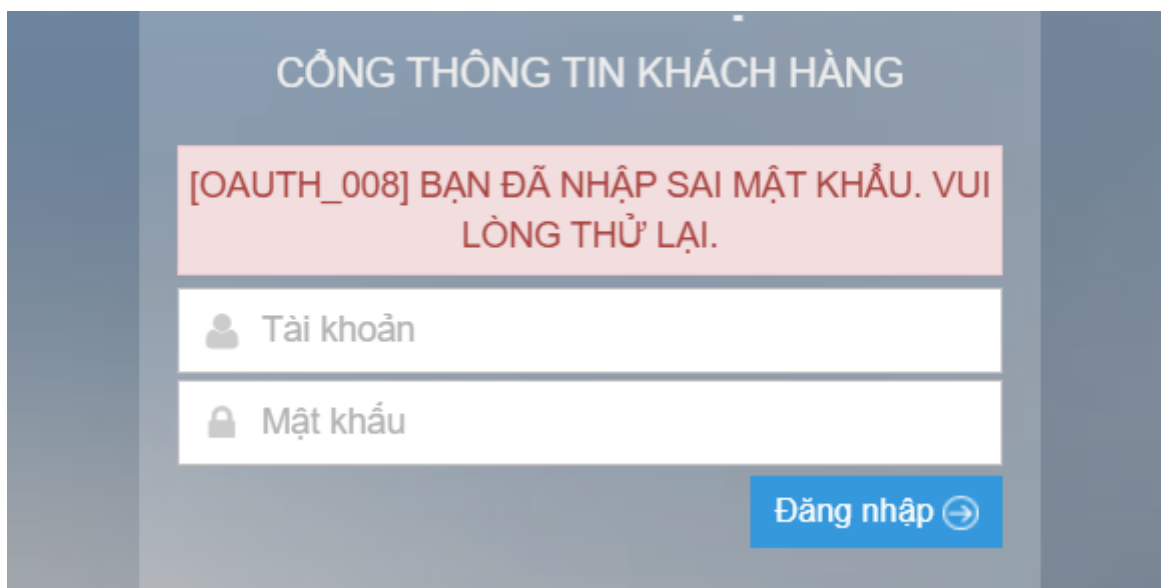


The screenshot shows the VETC login interface. At the top is the VETC logo. Below it, the text "ĐĂNG NHẬP" (Login) and "CỔNG THÔNG TIN KHÁCH HÀNG" (Customer Information Portal) are displayed. A red error message box states "TÀI KHOẢN & MẬT KHẨU KHÔNG ĐÚNG!" (Account & Password incorrect!). Below this are two input fields: "Tài khoản" (Account) and "Mật khẩu" (Password). A blue "Đăng nhập" (Login) button with a right arrow icon is positioned to the right of the password field. Below the button, there is a link "Quên mật khẩu ?" (Forgot password?) and a text "Click vào đây để lấy lại mật khẩu." (Click here to reset password.). At the bottom, there is a link "Tra cứu hóa đơn điện tử" (Check electronic invoice).

Hình 12. Đăng nhập tài khoản ngẫu nhiên

Khi đăng nhập không đúng tài khoản và mật khẩu thì sẽ có thông báo “TÀI KHOẢN & MẬT KHẨU KHÔNG ĐÚNG!”

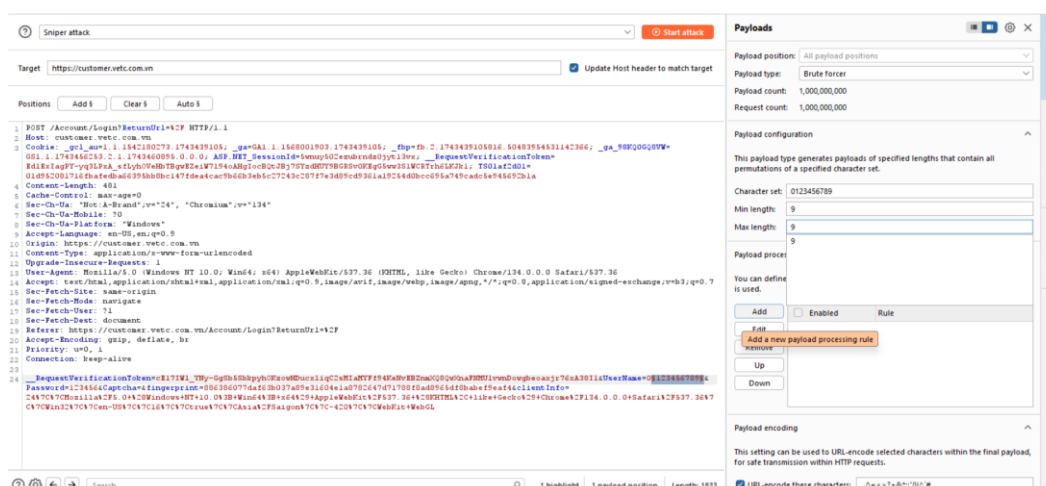
Đăng nhập với tên tài khoản đúng không quan tâm mật khẩu:



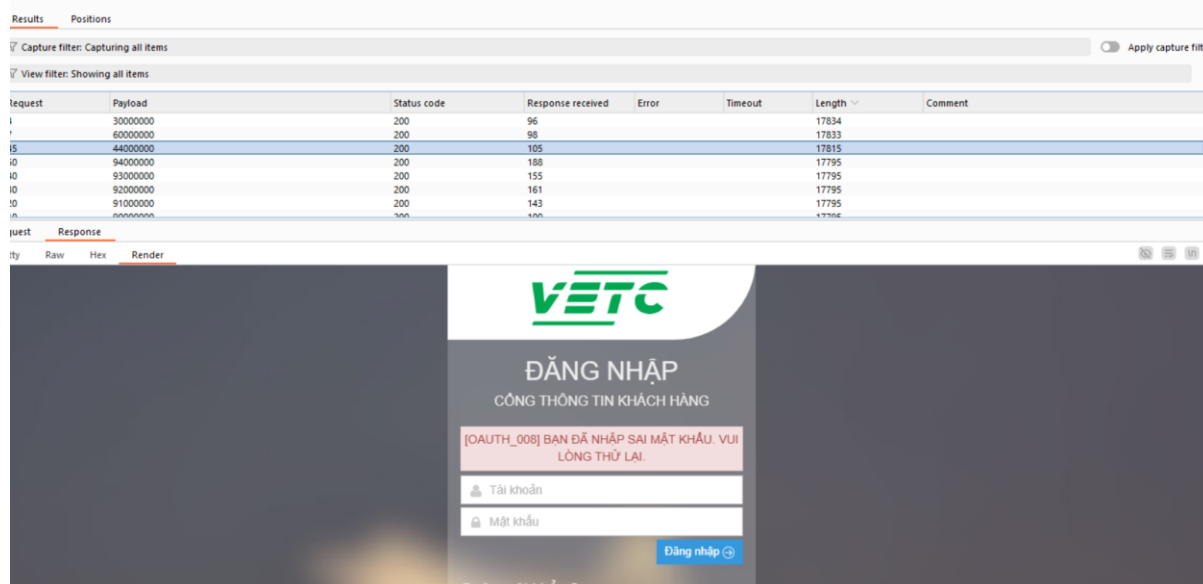
The screenshot shows the VETC login interface. At the top is the text "CỔNG THÔNG TIN KHÁCH HÀNG" (Customer Information Portal). Below it, a red message box states "[OAUTH\_008] BẠN ĐÃ NHẬP SAI MẬT KHẨU. VUI LÒNG THỬ LẠI." ([OAUTH\_008] You have entered the wrong password. Please try again.). Below this are two input fields: "Tài khoản" (Account) and "Mật khẩu" (Password). A blue "Đăng nhập" (Login) button with a right arrow icon is positioned to the right of the password field.

Hình 13. Đăng nhập đúng tài khoản

Ta tiến hành Brute Force để xem các số tài khoản tồn tại ở trang web:



Hình 14. Brute force



Hình 15: Thu thập các số điện thoại

Như vậy ta có thể thu thập khá nhiều số điện thoại của khách hàng

Biện pháp phòng ngừa:

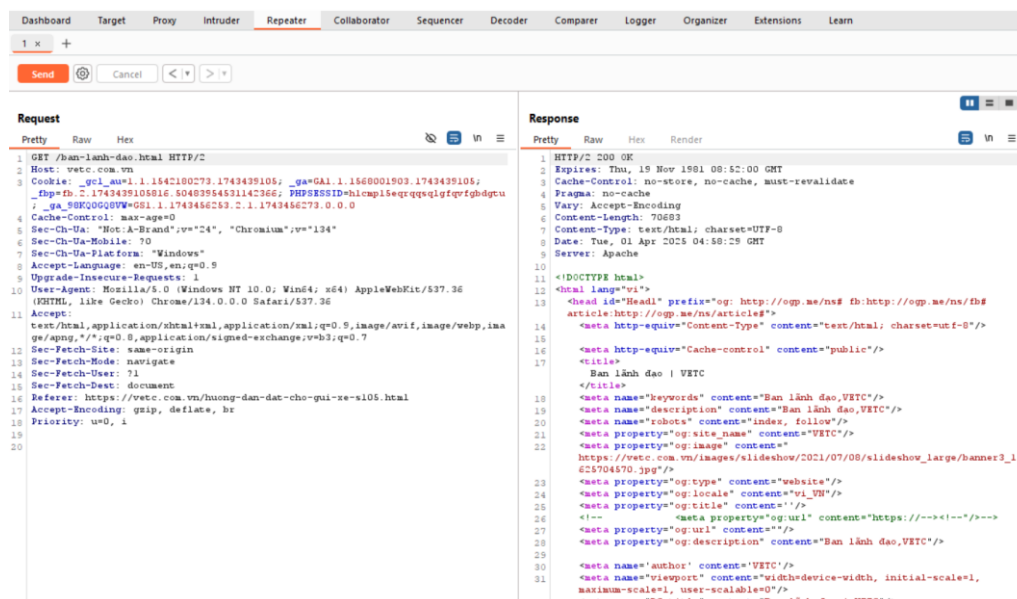
- Áp dụng nguyên tắc "Least Privilege": Chỉ cấp quyền tối thiểu cần thiết cho người dùng.
- Kiểm tra phân quyền: Đảm bảo người dùng không thể truy cập tài nguyên của người khác (vd: /user/profile?id=123).
- Sử dụng RBAC/ABAC: Quản lý quyền bằng Role-Based Access Control hoặc Attribute-Based Access Control.

- Mặc định từ chối (Deny-by-default): Chặn tất cả truy cập trừ khi được cho phép rõ ràng.
- Kiểm thử API: Đảm bảo API không bị lộ dữ liệu do thiếu kiểm tra quyền (vd: API trả về dữ liệu nhạy cảm mà không xác thực).

### 3.3. Kiểm thử A03: Injection

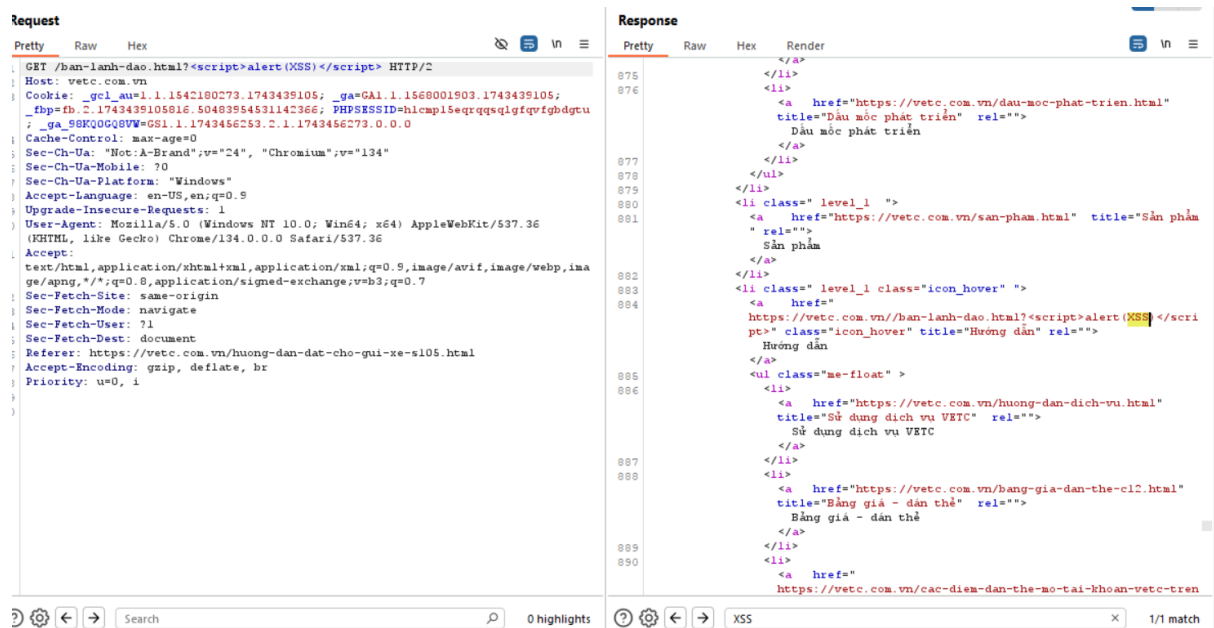
Ta chạy trang web với đường dẫn bắt kì: <https://vetc.com.vn/ban-lanh-dao.html> với BurpSuite

Và bắt gói tin truy cập trang web và gửi đến Repeater:



Hình 166. Request của đường dẫn

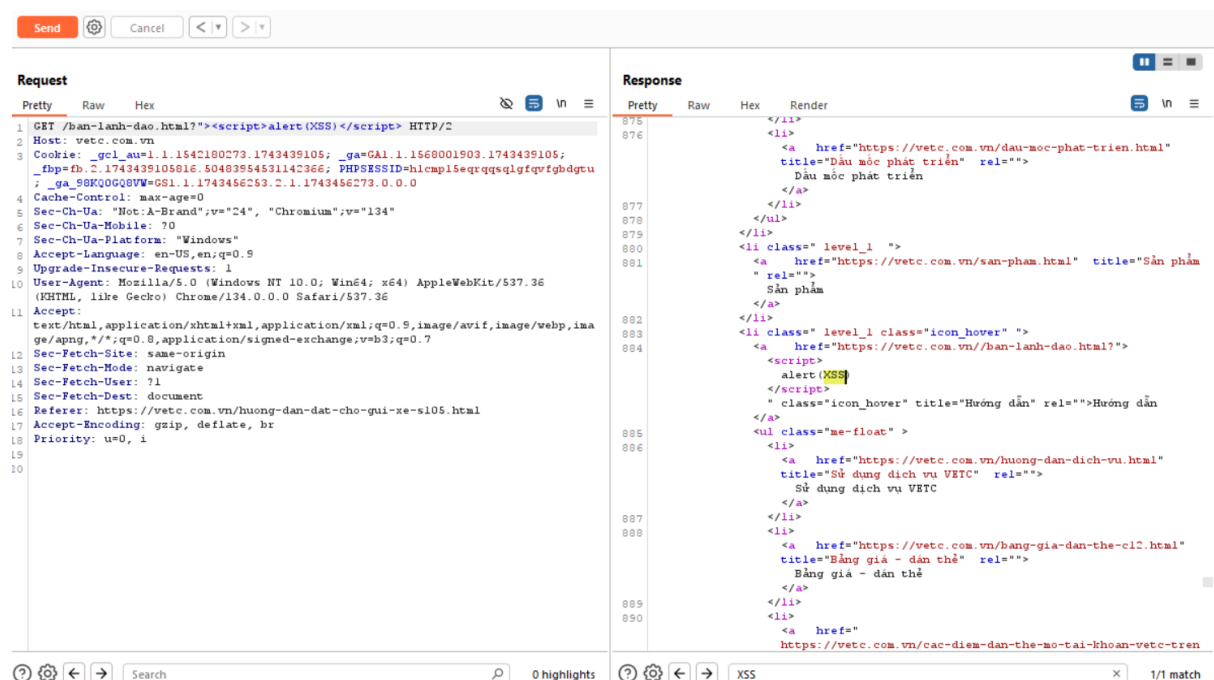
Thao tác thêm một đoạn mã để xác định lỗi XSS: “`<script>alert(XSS)</script>`”



Hình 17. Thêm đoạn mã XSS

Có thể thấy trang Web có thể bị tấn công XSS

Tiếp tục thay đổi đoạn mã, thêm “<” để định dạng lại thẻ <a>



Hình 18. Định dạng lại

- Nguy hiểm của XSS
  - o Đánh cắp thông tin nhạy cảm: Cookie, session, thông tin đăng nhập, dữ liệu cá nhân.

- Chiếm quyền điều khiển tài khoản: Kẻ tấn công có thể giả mạo người dùng, thực hiện các hành động như chuyển tiền, đăng bài, mua hàng.
- Phát tán mã độc: Redirect người dùng đến trang web lừa đảo (phishing) hoặc tải phần mềm độc hại.
- Thao túng giao diện: Thay đổi nội dung trang web, hiển thị quảng cáo giả mạo.
- Tấn công kết hợp với CSRF: Thực hiện các yêu cầu HTTP độc hại từ phía người dùng.

### Cách phòng chống XSS

- Đối với lập trình viên (phía server)
  - Validate & Sanitize Input:
    - Kiểm tra dữ liệu đầu vào (ví dụ: email phải đúng định dạng, không chứa thẻ HTML).
    - Sử dụng thư viện như DOMPurify (JavaScript) hoặc htmlspecialchars() (PHP) để làm sạch dữ liệu:
      - `echo htmlspecialchars($user_input, ENT_QUOTES, 'UTF-8');`
  - Sử dụng CSP (Content Security Policy):
    - Giới hạn nguồn tải script, ngăn chặn mã độc từ bên ngoài.
      - `Content-Security-Policy: default-src 'self'; script-src 'self' https://trusted.cdn.com;`
  - HTTP-only & Secure Cookies:
    - Đánh dấu cookie với HttpOnly để ngăn JavaScript truy cập.
      - `setcookie("sessionID", "123", httponly=True, secure=True);`
  - Mã hóa dữ liệu đầu ra:
    - Sử dụng hàm escape phù hợp với ngữ cảnh (HTML, JavaScript, URL).

## 3.4. Kiểm thử A04: Insecure Design

Biện pháp phòng ngừa:

- Áp dụng Secure by Design
  - Threat Modeling (Mô hình hóa mối đe dọa): Sử dụng công cụ như Microsoft Threat Modeling Tool hoặc OWASP Threat Dragon để phân tích rủi ro ngay từ giai đoạn thiết kế.



- Security Requirements từ đầu: Đưa yêu cầu bảo mật vào tài liệu thiết kế (ví dụ: "Mọi API phải có xác thực JWT + RBAC").
- Thiết kế Defense in Depth
  - Kiểm soát truy cập mặc định "Từ chối": Mọi request phải được xác thực trừ khi công khai rõ ràng.
  - Zero Trust Architecture: Không tin cậy bất kỳ thành phần nào (kể cả nội bộ), luôn xác minh.
- Thiết kế cho tính bảo mật
  - Không lưu trữ dữ liệu nhạy cảm không cần thiết:
  - Tách biệt microservices: Dịch vụ xử lý thanh toán nên tách biệt với dịch vụ đăng nhập để giảm blast radius.
- Sử dụng Patterns an toàn
  - Cơ chế xác thực mạnh:
    - MFA (2FA) ngay từ thiết kế, không phải bổ sung sau.
  - API Gateway với Rate Limiting: Giới hạn 100 requests/phút cho API đăng nhập để chống brute force.

### 3.5. Kiểm thử A05: Security Misconfiguration

#### Biện Pháp Phòng Ngừa & Khắc Phục

- Loại Bỏ Cấu Hình Mặc Định Nguy Hiểm
  - Thay đổi mật khẩu mặc định của hệ thống, database, ứng dụng.
  - Tắt các dịch vụ không cần thiết (VD: FTP, Telnet, debug mode trong production).
  - Xóa tài khoản mẫu (VD: admin/admin, guest/guest).
- Cập Nhật & Patch Management
  - Luôn cập nhật:
    - Hệ điều hành (OS updates).
    - Web server (Apache, Nginx), database (MySQL, PostgreSQL).
    - Framework (Laravel, Django, Spring) và thư viện (npm, pip).
- Sử dụng công cụ quét lỗ hổng:
  - Trước khi triển khai: SCA (Software Composition Analysis) như Snyk, Dependabot.
  - Sau triển khai: Nessus, OpenVAS, Trivy (quét container).

- Kiểm Soát Quyền Truy Cập (Permissions)
  - o File/Folder Permissions:
    - Không để file nhạy cảm (config.yml, .env, backup.sql) trong thư mục web root.
    - Đặt quyền chmod 600 cho file chứa mật khẩu, 750 cho thư mục.
  - o Database Permissions:
    - o Giới hạn quyền của DB user (VD: SELECT thay vì ALL PRIVILEGES).
- Cấu Hình Headers & HTTPS An Toàn
  - o HTTP Security Headers:
 

```
# Nginx config

add_header X-Frame-Options "DENY";

add_header X-Content-Type-Options "nosniff";

add_header Content-Security-Policy "default-src 'self'";

add_header Strict-Transport-Security "max-age=63072000;
includeSubDomains; preload";
```
  - o Luôn dùng HTTPS (không HTTP):
    - Cấu hình redirect HTTP → HTTPS.
    - Sử dụng Let's Encrypt (miễn phí) hoặc mua SSL/TLS từ nhà cung cấp uy tín.
- Giới Hạn Thông Báo Lỗi
  - o Tắt debug mode trong production:
    - PHP: display\_errors = Off (trong php.ini).
    - Django: DEBUG = False (trong settings.py).
    - Spring Boot: server.error.include-stacktrace=never.
  - o Custom error pages để tránh lộ thông tin server.
- Kiểm Tra CORS & API Security
  - o CORS (Cross-Origin Resource Sharing):
 

```
# Chỉ cho phép domain cụ thể

add_header 'Access-Control-Allow-Origin' 'https://trusted-site.com';

add_header 'Access-Control-Allow-Methods' 'GET, POST';
```
  - o API Security:
    - Không dùng API key trong URL (?api\_key=xxx), thay bằng Authorization: Bearer.
    - Giới hạn rate limiting (VD: 100 requests/phút).

- Sử dụng Security Hardening Guides
  - o OWASP Secure Configuration Guide: <https://cheatsheetseries.owasp.org/>
  - o CIS Benchmarks: <https://www.cisecurity.org/cis-benchmarks/> (cấu hình an toàn cho OS, DB, cloud)

### 3.6. Kiểm thử A06: Vulnerable Components

- Lý Dependencies Chặt Chẽ
  - o Sử dụng công cụ quét lỗ hổng tự động:
    - SCA (Software Composition Analysis):
      - Snyc (<https://snyk.io/>)
      - (tích hợp với GitHub)
      - Dependency-Check (<https://owasp.org/www-project-dependency-check/>)
  - o Ghim phiên bản thư viện (version pinning):

Trong package.json (Node.js):

```
"dependencies": {
  "express": "4.17.3" # Không dùng "^4.17.3" (tránh tự động cập nhật gây break)
}
```

- Cập Nhật Thường Xuyên
  - o Thiết lập quy trình cập nhật định kỳ:
    - Hàng tuần/tháng kiểm tra bản vá mới.
    - Sử dụng GitHub Dependabot hoặc RenovateBot để tự động đề xuất updates.
  - o Ưu tiên nâng cấp các thành phần nguy hiểm:
    - Framework (Spring, Django, Laravel).
    - Thư viện mã hóa (OpenSSL, bcrypt).
    - Công cụ logging (Log4j, Winston).
- Giảm Thiểu Sử Dụng Thành Phần Không Cần Thiết
  - o Loại bỏ các dependencies không dùng đến:
 

```
npm prune # Xóa packages không sử dụng trong Node.js
```
  - o Tránh kéo cả "kitchen sink" (VD: Import cả thư viện lớn như lodash trong khi chỉ cần 1-2 hàm).
- Theo Dõi CVE & Bản Tin Bảo Mật

- Đăng ký nhận thông báo lỗ hổng:
  - CVE Database: <https://cve.mitre.org/>
  - National Vulnerability Database (NVD): <https://nvd.nist.gov/>
  - Security mailing lists (VD: OWASP, Full Disclosure).
- Sử Dụng SBOM (Software Bill of Materials)
  - SBOM giúp liệt kê tất cả thành phần trong ứng dụng, dễ dàng truy vết lỗ hổng.
  - Công cụ tạo SBOM:
    - Syft: <https://github.com/anchore/syft>
    - SPDX: <https://spdx.dev/>

## CHƯƠNG 4: KẾT LUẬN

### 4.1. Đánh giá Mức độ An toàn của Trang Web

Trang web đang có mức an toàn 30%, cho thấy nhiều rủi ro bảo mật nghiêm trọng, đặc biệt liên quan đến các hạng mục trong OWASP Top 10 (2021) như:

A01 – Broken Access Control: Kiểm soát truy cập bị lỗi có thể cho phép kẻ tấn công truy cập trái phép vào dữ liệu hoặc chức năng nhạy cảm, gây rò rỉ thông tin hoặc thay đổi dữ liệu quan trọng.

A03 – Injection: Trang web có thể dễ bị tấn công bằng các phương pháp như SQL Injection, Command Injection hoặc Cross-Site Scripting (XSS), cho phép kẻ tấn công thực thi mã độc hoặc truy xuất dữ liệu trái phép.

A04 – Insecure Design: Thiết kế bảo mật không đầy đủ có thể dẫn đến việc thiếu các biện pháp bảo vệ dữ liệu quan trọng, làm tăng nguy cơ bị khai thác.

A05 – Security Misconfiguration: Các cấu hình bảo mật yếu như mở port không cần thiết, lộ thông tin debug, cấu hình mặc định có thể giúp hacker dễ dàng khai thác hệ thống.

A06 – Vulnerable and Outdated Components: Sử dụng các thư viện, framework hoặc phần mềm cũ không được cập nhật có thể chứa lỗ hổng bảo mật đã biết, tạo điều kiện cho tấn công từ xa.

### 4.2. Khuyến nghị cải thiện

Kiểm tra và áp dụng chính sách kiểm soát truy cập chặt chẽ (A01), hạn chế quyền không cần thiết.

Kiểm tra và khắc phục các lỗ hổng Injection (A03) bằng cách sử dụng prepared statements và lọc dữ liệu đầu vào.

Cải thiện thiết kế bảo mật (A04) bằng cách áp dụng Secure by Design, kiểm tra logic kinh doanh để tránh bị khai thác.

Kiểm tra và sửa đổi cấu hình bảo mật (A05), đảm bảo máy chủ và ứng dụng không lộ thông tin nhạy cảm.

Cập nhật các thành phần lỗi thời (A06), sử dụng các bản vá bảo mật mới nhất để tránh khai thác từ lỗ hổng cũ.

## TÀI LIỆU THAM KHẢO