|  |  |
| --- | --- |
| **BÁO CÁO**  ***PENETRATION TESTING***  Website Đặng Hùng Vĩ  **http://172.16.88.30:9999/web\_ban\_hang** |  |

**Phần 1: Kết quả đánh giá**

|  |  |
| --- | --- |
| **Thời gian đánh giá** | Tháng 08/2024 |
| **Kết quả chung** | Website đang có lỗ hổng liên quan tới bussiness logic, CSRF |
| **Đề xuất** | Khắc phục các lỗi được đưa ra trong báo cáo. Về dài hạn, cần triển khai đồng bộ các giải pháp ATTT nhằm phát hiện, cảnh báo các hành vi bất thường khi truy cập vào hệ thống, đồng thời triển khai việc theo dõi các lỗ hổng, nguy cơ mới để từ đó có phương án nâng cấp, giảm thiểu các rủi ro này. |
| **Số lượng lỗi nghiêm trọng** | 1 |
| **Số lượng lỗi cao** | 2 |
| **Số lượng lỗi trung bình** | 1 |
| **Số lượng lỗi thấp** | 0 |
| **Ghi chú (nếu có)** |  |

**Phần 2: Thông tin về các lỗ hổng**

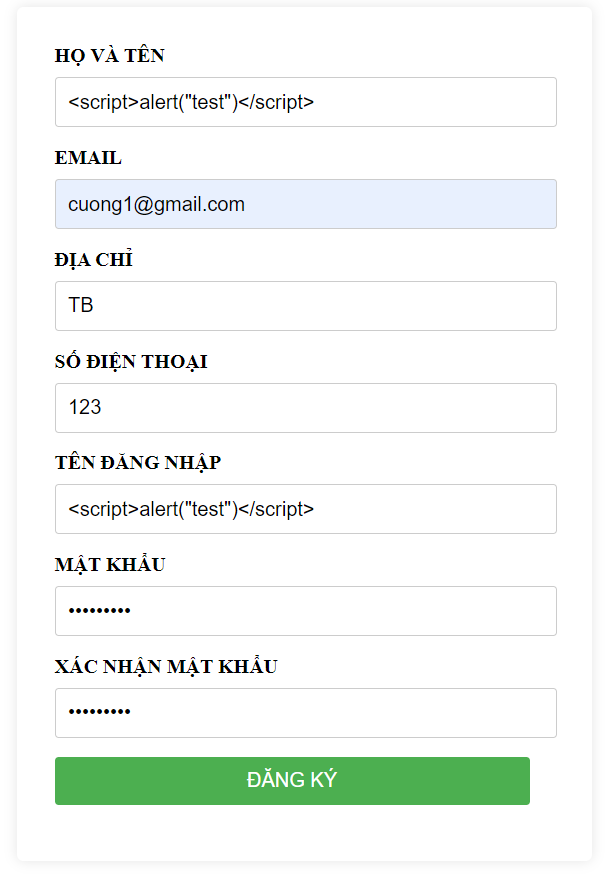
1. **Danh sách lỗ hổng**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Lỗi** | **Mô tả** | **Mức độ ảnh hưởng** |
| 1 | Lỗ hổng Stored-XSS ở username | Trong quá trình phân tích trang web, tài khoản có thể đăng ký với các ký tự đặc biệt, điều này gây ra lỗ hổng XSS, trang quản lý người dùng của admin nếu xử lý không đúng cách, kẻ tấn công có thể đánh cắp phiên của quản trị để truy cập với tài khoản quản trị | Cao |
| 2 | Lỗ hổng business logic đặt hàng với giá tiền âm | Request thêm giỏ hàng hiện rõ các thuộc tính giá tiền,… Kẻ tấn công có thể bắt request này để thực hiện việc đặt hàng mà không mất tiền, nếu admin không kiểm tra kỹ mà xác nhận đơn hàng thì sẽ làm thiệt hại doanh thu của cửa hàng | Nghiêm trọng |
| 3 | Lỗ hổng business logic có thể bình luận sản phẩm khi chưa mua hàng | Request addComment hiện rõ thuộc tính mã sản phẩm và nội dung bình luận, kẻ tấn công có thể bắt request để sửa bình luận các sản phẩm khi chưa mua hàng, có thể làm giảm uy tín của cửa hàng | Cao |
| 4 | Lỗ hổng CSRF tự động thêm giỏ hàng | Khi phân tích chức năng thêm sản phẩm vào giỏ hàng nhận thấy request này thoả mãn 1 cuộc tấn công CSRF, kẻ tấn công lợi dụng lỗ hổng này để lừa người dùng thêm giỏ hàng làm giảm trải nghiệm người dùng | Trung Bình |

1. **Chi tiết lỗ hổng**
2. **Lỗ hổng Stored-XSS ở username**

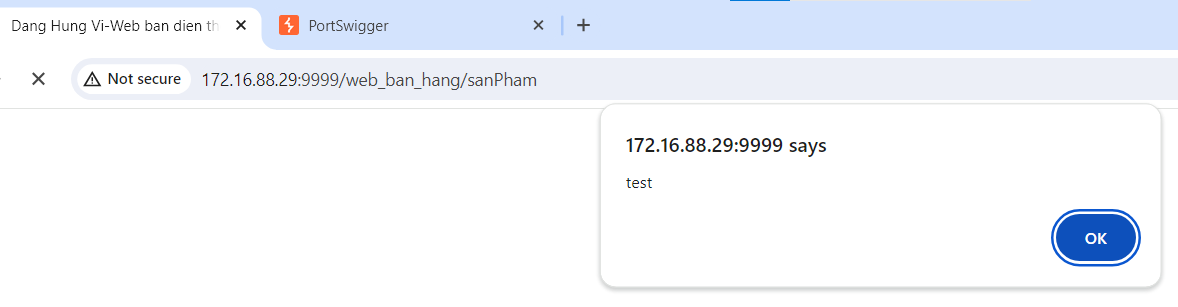
* **Mức độ: Cao**
* **Mô tả:** Khi đăng ký 1 tài khoản username có dạng <script>…</script>, có thể gây ra lỗi Stored-XSS khi comment, hoặc gây ra Stored-XSS ở trang quản trị

Đăng ký tài khoản



Hình 1: Đăng ký tài khoản bị lỗi XSS

Lỗi XSS này có thể gây ra Stored-XSS ở comment, hoặc ở trang quản lý tài khoản của admin nếu không kiểm tra đầu vào đúng cách.



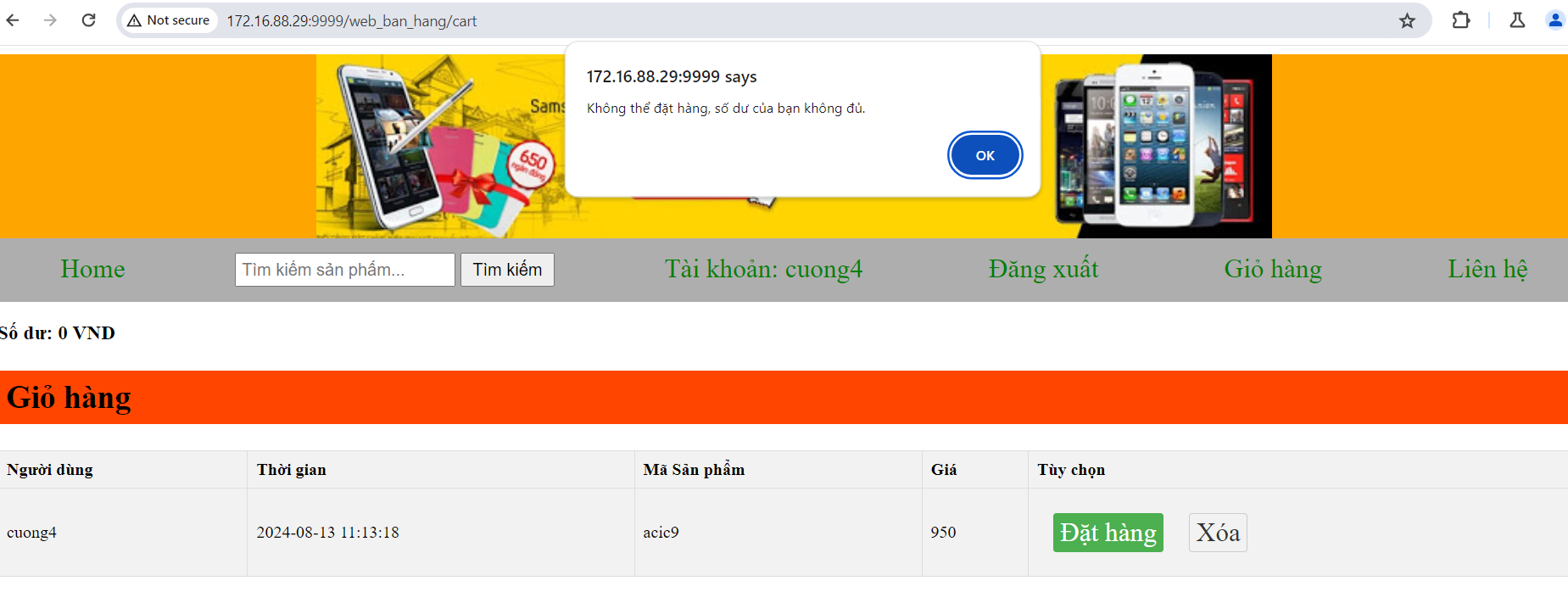
Hình 2: Trang web đã bị lỗi XSS

* **Khuyến nghị:**
  + Xác thực và làm sạch đầu vào: Sử dụng các thư viện hoặc hàm chuẩn để loại bỏ hoặc mã hóa các ký tự đặc biệt như <, >, ", ', và & để ngăn chặn chèn mã độc.
  + Mã hóa đầu ra: Mã hóa tất cả các dữ liệu người dùng trước khi hiển thị lên trang web (output encoding). Điều này bao gồm cả dữ liệu được hiển thị trong các thẻ HTML, thuộc tính của thẻ, và JavaScript.

1. **Lỗ hổng business logic đặt hàng với giá tiền âm**

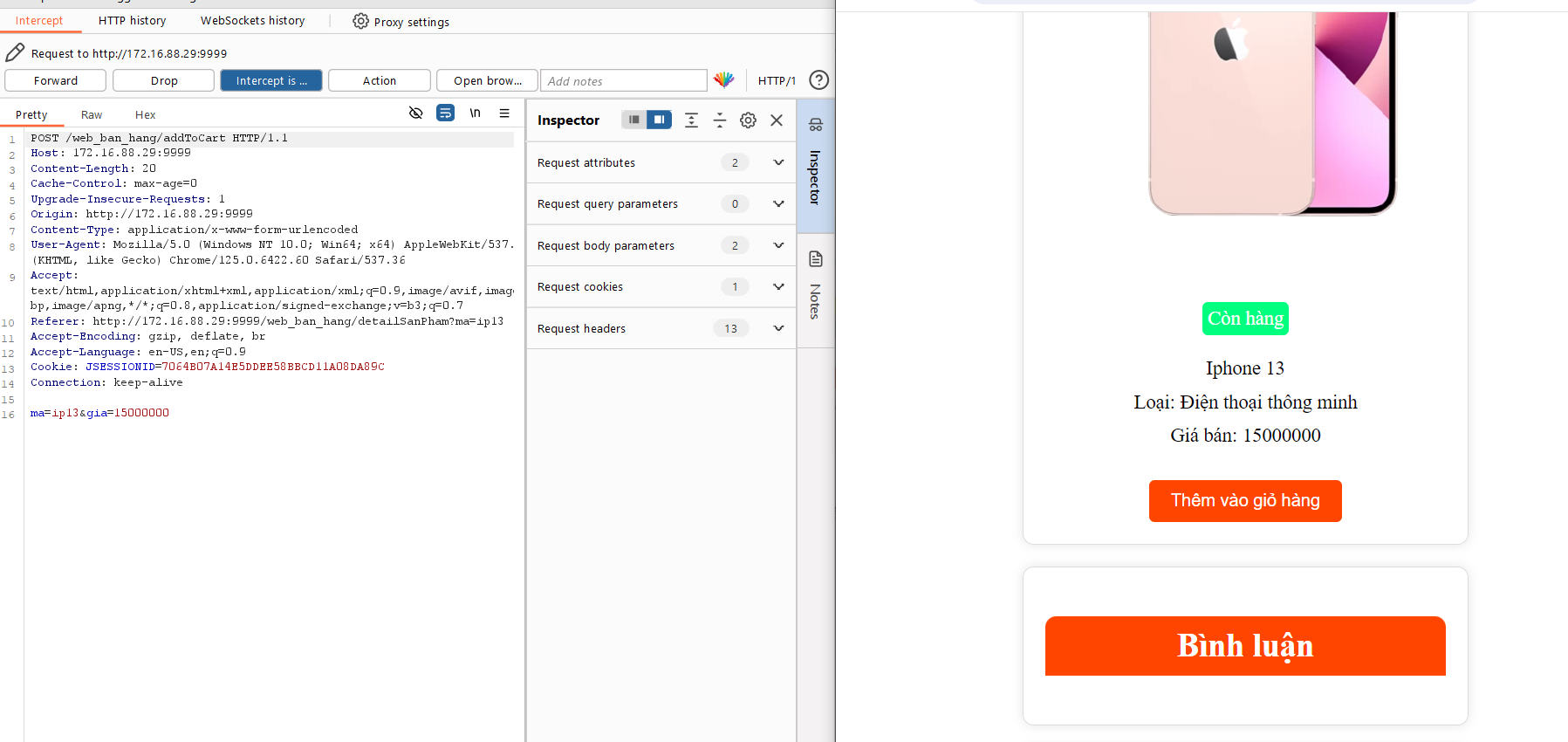
* **Mức độ: Nghiêm trọng**
* **Mô tả:** Khi đặt hàng với đơn hàng có số tiền lớn hơn số dư trong tài khoản thì không thể đặt hàng được, kẻ tấn công có thể bắt sự kiện thêm giỏ hàng để sửa giá tiền của sản phẩm sau đó đặt hàng sản phẩm đó.

Nếu đặt hàng mà chưa có tiền trong tài khoản thì không thể đặt hàng được



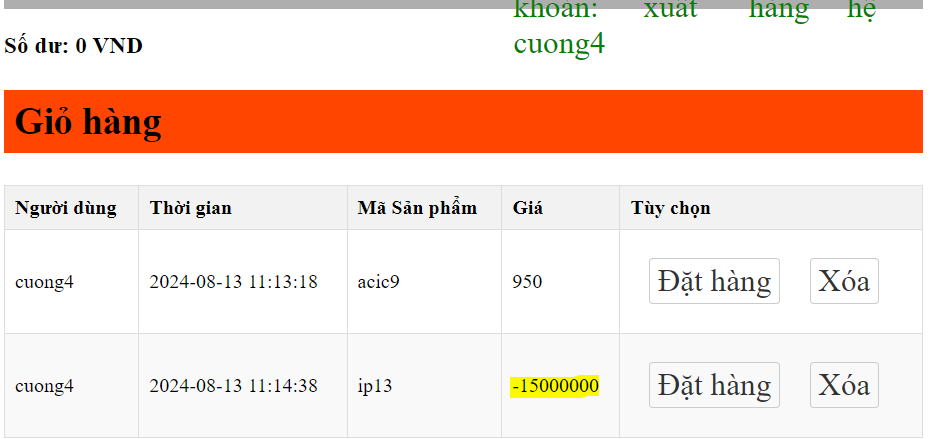
Hình 3: Tài khoản có số dư 0 VND không thể đặt hàng được

Bật Intercept bắt sự kiện nút thêm giỏ hàng



Hình 4: Bắt sự kiện nút thêm giỏ hàng

Sửa thành giá tiền âm sau đó Forward 2 lần



Hình 5: Số tiền của đơn hàng “ip13” đã thay đổi trong giỏ hàng

Nhấn đặt hàng, mặc dù Số dư 0 VND nhưng vẫn có thể đặt hàng được



Hình 6: Tài khoản cuong4 đã đặt hàng thành công mặc dù có số dư 0 VND

Nếu admin không check kỹ xác nhận đơn hàng, số dư trong tài khoản cũng tự động cập nhật lên



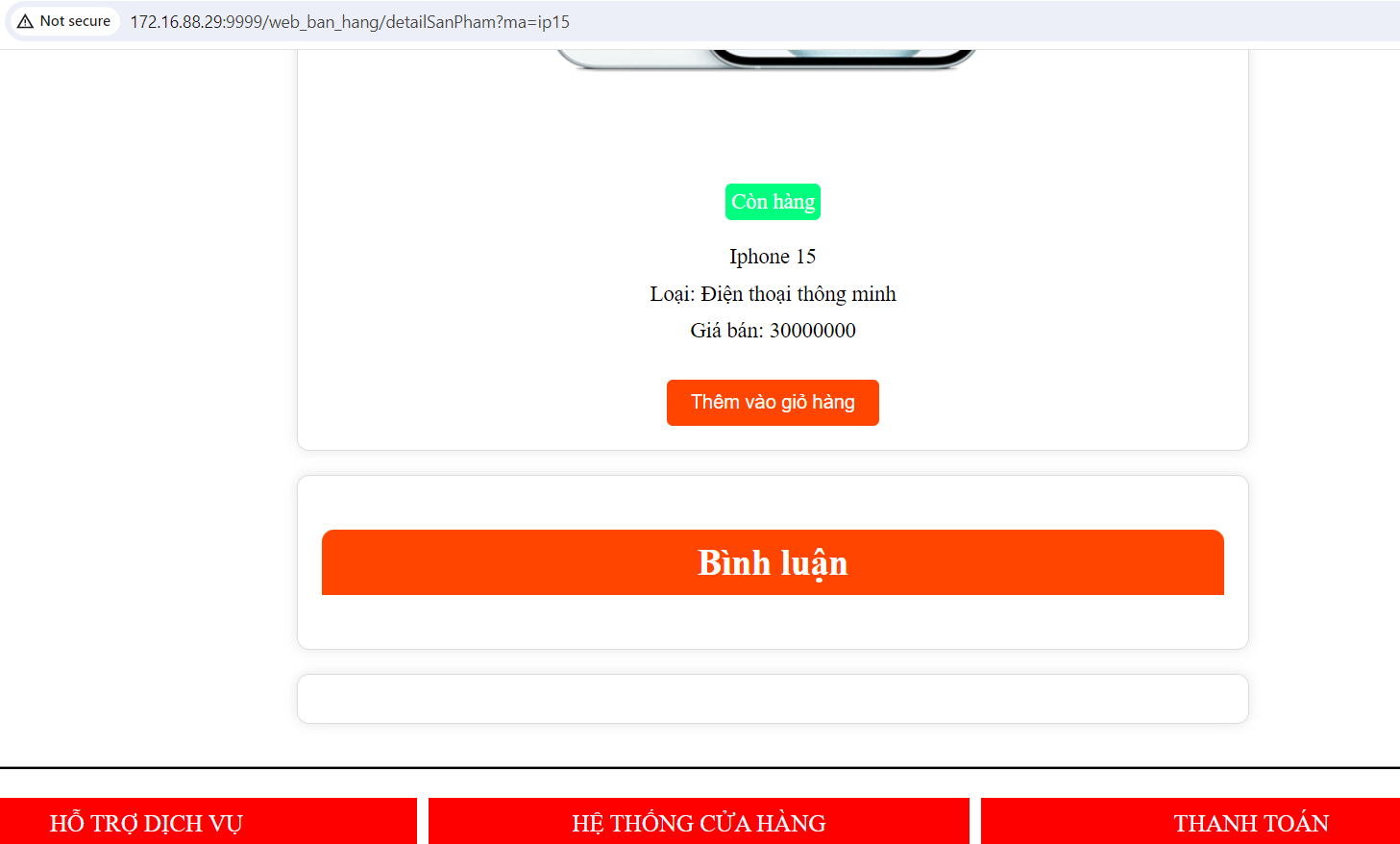
Hình 7: Số dư tài khoản cuong4 đã tăng lên

* **Khuyến nghị:**
  + Xác thực và kiểm tra dữ liệu phía server: Không dựa vào dữ liệu từ phía client để tính toán giá tiền cuối cùng của đơn hàng. Mọi phép tính liên quan đến giá tiền cần được thực hiện trên server, và giá trị cuối cùng cần được xác nhận trước khi xử lý thanh toán.
  + Giới hạn và kiểm soát đầu vào: Áp dụng giới hạn cho tất cả các trường số đầu vào liên quan đến số lượng sản phẩm và giá tiền.
  + Giám sát và cảnh báo: Thiết lập các hệ thống giám sát để phát hiện và cảnh báo về các hoạt động bất thường, như việc cố gắng đặt hàng với giá trị âm hoặc sửa đổi bất hợp pháp dữ liệu giao dịch.

1. **Lỗ hổng business logic có thể bình luận sản phẩm khi chưa mua hàng**

* **Mức độ: Cao**
* **Mô tả:** Khi chưa mua hàng thì người dùng chưa thể bình luận ở sản phẩm đó được, nhưng kể tấn công có thể sửa request bình luận ở sản phẩm đã mua để bình luận vào sản phẩm chưa mua

Khi chưa mua hàng thì chưa thể bình luận được



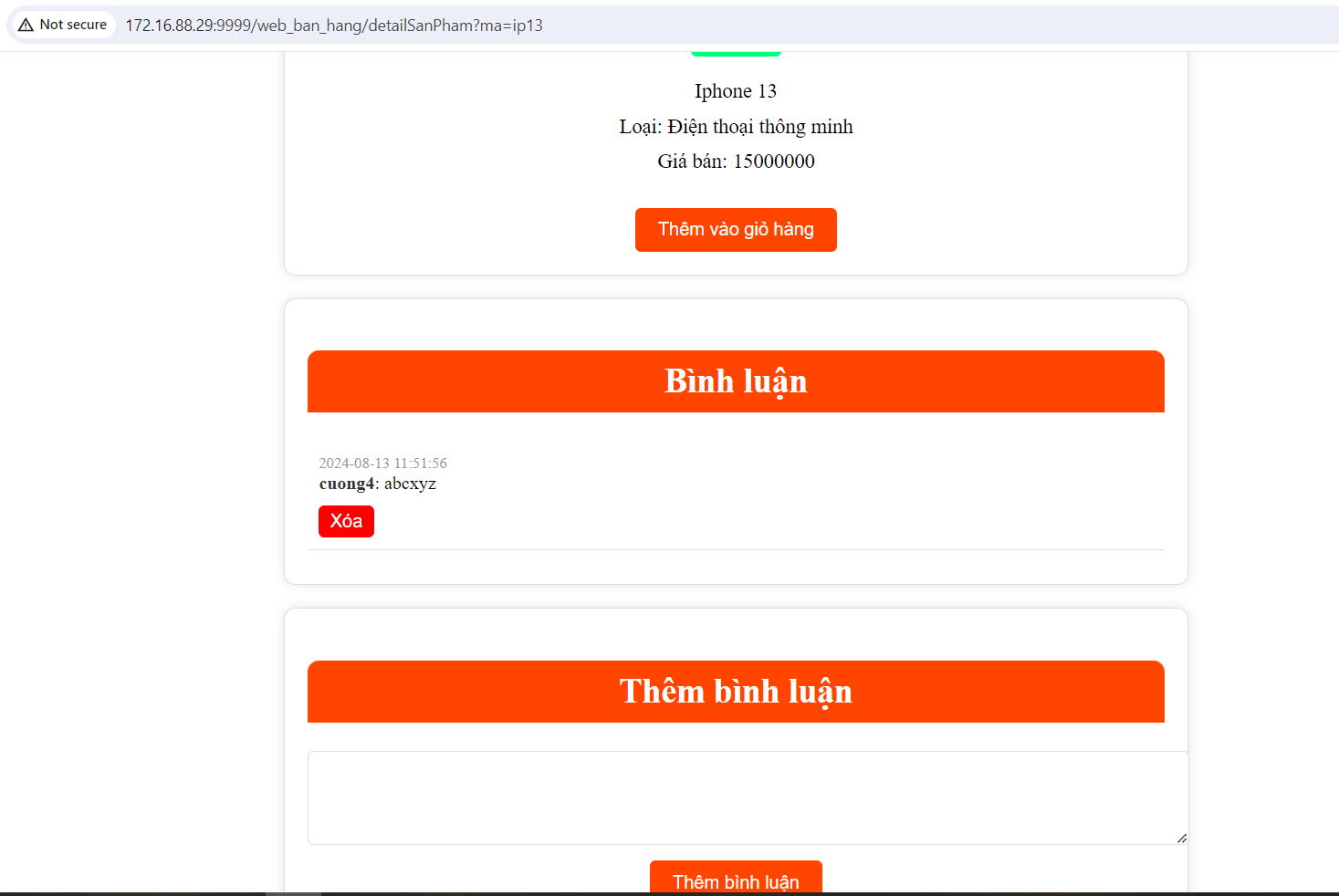
Hình 8: Chưa mua hàng thì chưa bình luận được

Đây là sản phẩm mà người dùng đã mua



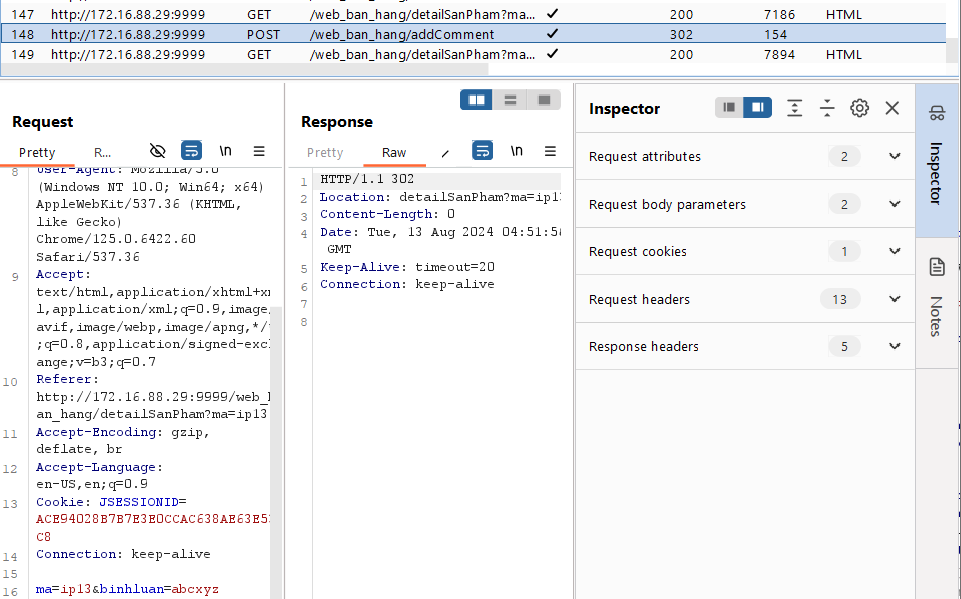
Hình 9: Tài khoản cuong4 đã đặt 1 sản phẩm có mã “ip13”

Sản phẩm đã mua thì có thể bình luận được



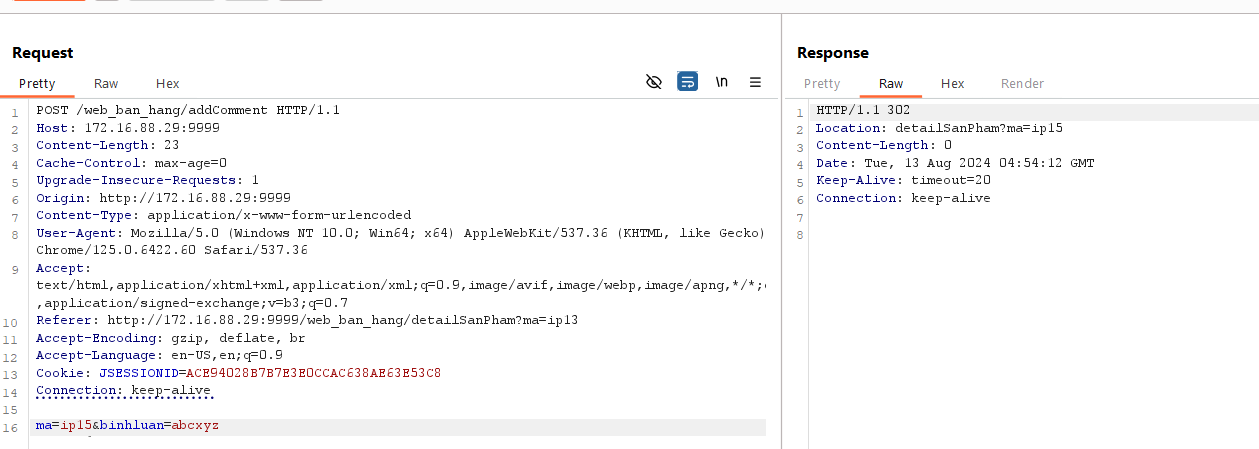
Hình 10: Cuong4 có thể bình luận sản phẩm có mã “ip13”

Bắt sự kiện thêm bình luận sản phẩm mã “ip13” gửi sang Repeater



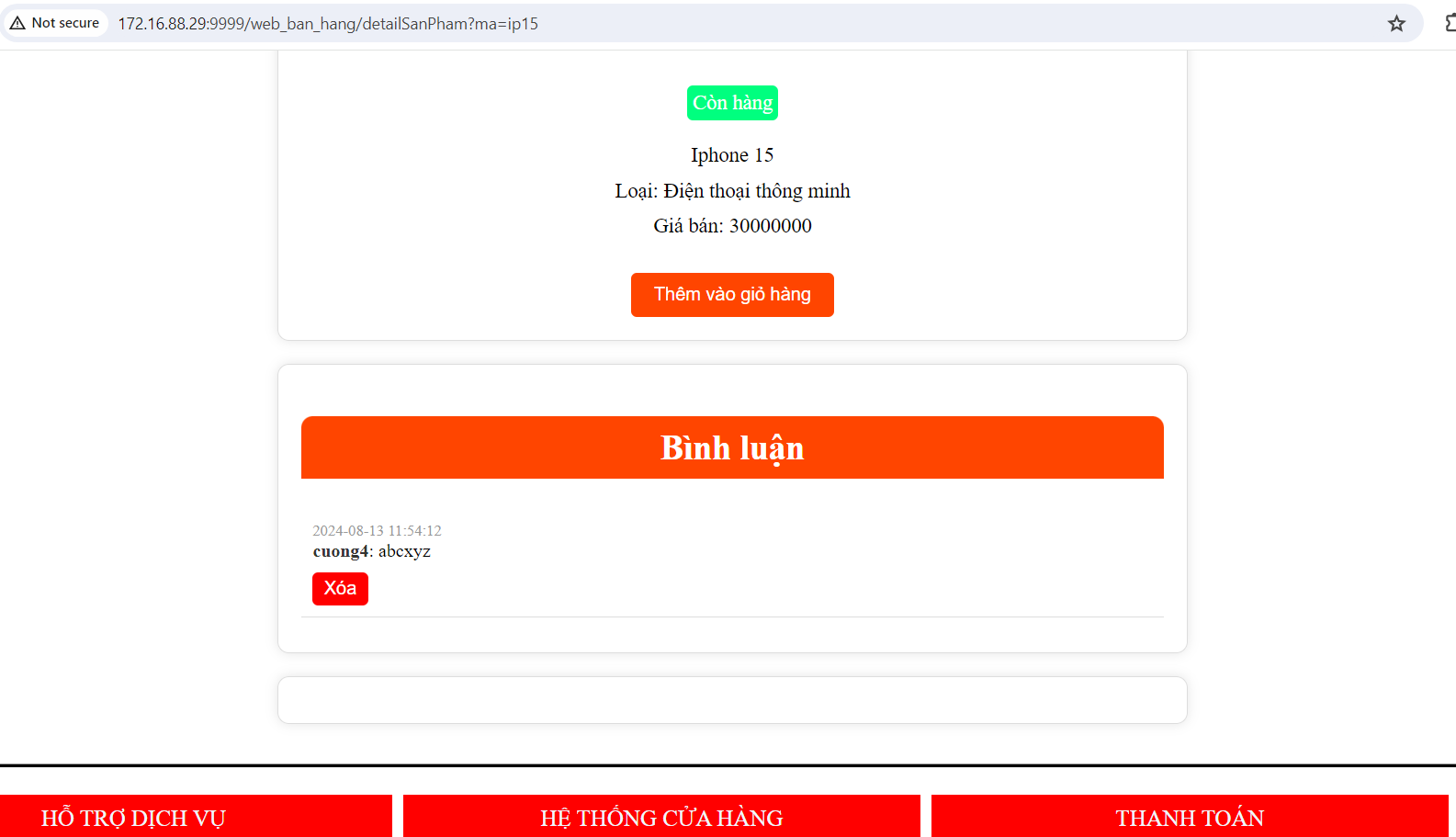
Hình 11S: Request thêm bình luận

Sửa mã ip13 thành mã của sản phẩm ip15 trên sau đó Send



*Hình 12. Sửa ma=ip13 thành ma=ip15 sau đó Send lại*

Vào trang của sản phẩm mã “ip15” đã hiện bình luận

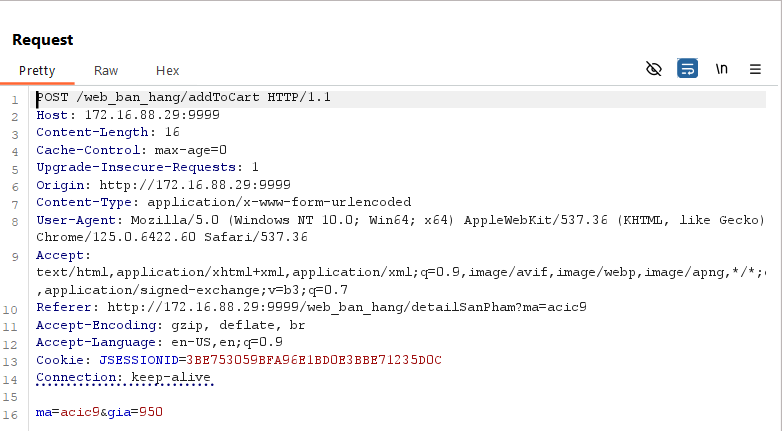


*Hình 13. Cuong4 chưa mua sản phẩm có mã “ip15” nhưng vẫn có thể bình luận được*

* **Khuyến nghị:**
  + Sử dụng mã hóa và kiểm tra tính toàn vẹn dữ liệu: Mã hóa các thông tin quan trọng trong request bình luận (như ID sản phẩm) để tránh việc sửa đổi dữ liệu từ phía client.

1. **Lỗ hổng CSRF tự động thêm sản phẩm vào giỏ hàng**

* **Mức độ: Trung Bình**
* **Mô tả:** Thêm 1 sản phẩm bất kỳ vào giỏ hàng, bắt request gửi sang Repeater để phân tích



Hình 14: Bắt request thêm sản phẩm có mã “acic9” và giá là 950 vào giỏ hàng

Phân tích request thoả mãn 3 điều kiện của 1 cuộc tấn công CSRF:

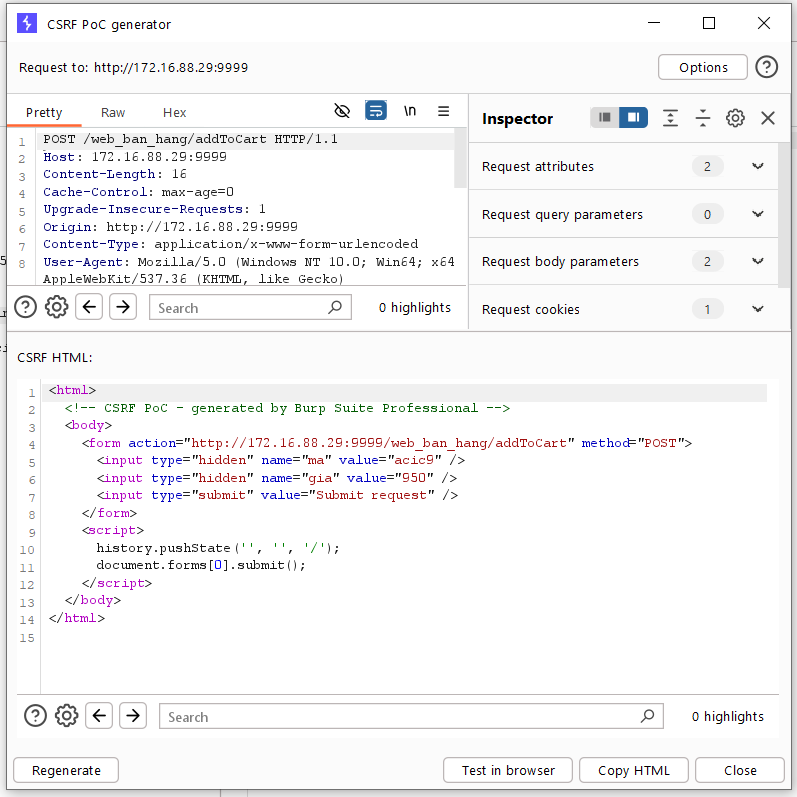
- Một hành động liên quan: Hành động thêm sản phẩm vào giỏ hàng.

- Xử lý phiên dựa trên cookie: Thoả mãn có Cookie: JSESSIONID=…

- Không có tham số yêu cầu không thể đoán trước: Thoả mãn, các tham số “ma”, “gia” có thể tìm trong shop.

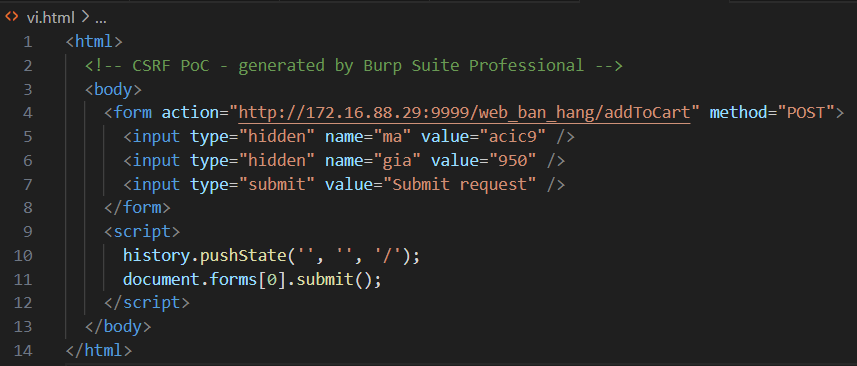
🡪 Thoả mãn để thực hiện 1 cuộc tấn công CSRF

Tạo yêu cầu độc hại

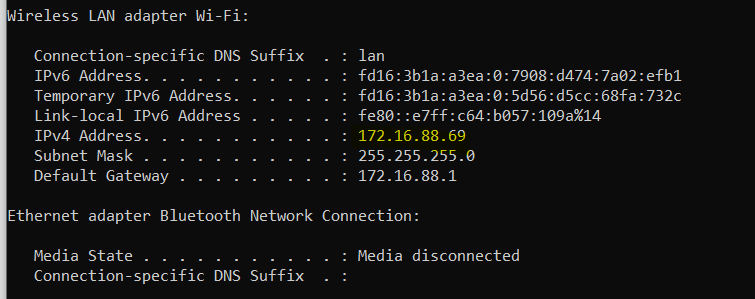


Hình 15: CSRF PoC generator

Chạy đoạn code trên



Hình 16: Code tạo yêu cầu độc hại

IP của máy tấn công: 

Hình 17: IP cúa máy tấn công là 172.16.88.69

Sau đó gửi link http://172.16.88.69:5500/vi.html cho nạn nhân, khi nạn nhân click link này, sản phẩm tự động thêm vào giỏ hàng



*Hình 18: Sau khi lừa người dùng click vào link thì sản phẩm acic9 tự động thêm vào giỏ hàng*

* **Khuyến nghị:**
  + Sử dụng mã token CSRF (CSRF Token): Áp dụng mã token CSRF cho tất cả các form và yêu cầu POST, đặc biệt là các hành động nhạy cảm như thêm sản phẩm vào giỏ hàng.
  + Kiểm tra và xác thực nguồn gốc yêu cầu (Referer Header Check): Xác thực rằng tất cả các yêu cầu liên quan đến giỏ hàng xuất phát từ cùng một miền (domain) bằng cách kiểm tra header Referer hoặc Origin. Nếu yêu cầu đến từ một nguồn không hợp lệ, yêu cầu đó cần bị từ chối.
  + Sử dụng SameSite Cookie: Cấu hình cookie phiên (session cookie) với thuộc tính SameSite để ngăn chặn các yêu cầu từ các trang web khác (cross-site request) mà không được phép. Điều này sẽ giúp giảm thiểu nguy cơ các cuộc tấn công CSRF.