|  |  |
| --- | --- |
| **BÁO CÁO**  ***PENETRATION TESTING***  Website Nguyễn Xuân Hiếu  **http://172.16.88.67:8080/MS/start** |  |

**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, XSS |
| **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** | 3 |
| **Số lượng lỗi cao** | 4 |
| **Số lượng lỗi trung bình** | 0 |
| **Số lượng lỗi thấp** | 0 |
| **Ghi chú (nếu có)** |  |

**Phần 2: Thông tin về các lỗ hổng**

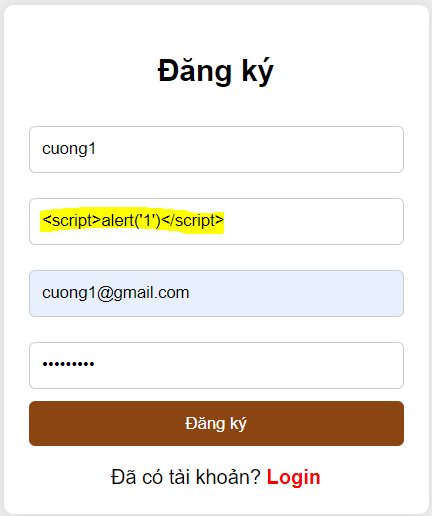
1. **Danh sách lỗ hổng**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên Lỗi** | **Mô tả** | **Mức độ ảnh hưởng** |
| 1 | Lỗ hổng XSS ở fullname | Bình luận sản phẩm, form bình luận có dạng (Tên người dùng: nội dung bình luận), khi đăng ký tên người dùng không được kiểm tra kỹ lưỡng dẫn đến xảy ra lỗi Stored-XSS, kẻ tấn công có thể lợi dụng để đánh cắp phiên người dùng khác | Cao |
| 2 | Lỗ hổng DOM XSS trong cập nhật thông tin tài khoản | Khi chỉnh sửa bất kỳ thông tin nào trong mục cập nhật thông tin tài khoản, trang web bị lỗi DOM-XSS. | Cao |
| 3 | Lỗ hổng business logic chỉnh sửa số lượng của sản phẩm về số âm trên trang web | Có thể chỉnh sửa trực tiếp số lượng sản phẩm ở trang web, tuy nhiên việc không kiểm tra đầu vào kỹ lưỡng dẫn đến việc có thể chỉnh sửa số lượng sản phẩm về giá trị âm, ảnh hưởng đến tài chính của cửa hàng | Nghiêm trọng |
| 4 | Lỗ hổng business logic mua hàng với giá tiền âm | Action /buy của request /MS/cart hiện rõ thông tin về trường “giá tiền”. Lợi dụng điều này kẻ tấn công có thể chỉnh sửa giá tiền của sản phẩm về giá tiền âm thực hiện mua hàng, gây thiệt hại về tài chính của cửa hàng. | Nghiêm trọng |
| 5 | Lỗ hổng CSRF trong cập nhật thông tin cá nhân | Request POST /MS/profile có thể lợi dụng để kẻ tấn công thực hiện 1 cuộc tấn công CSRF, khi người dùng không may truy cập link độc hại này, kẻ tấn công có thể đánh cắp tài khoản người dùng. | Cao |
| 6 | Lỗ hổng CSRF trong mua hàng ở giỏ hàng | Request POST MS/cart thoả mãn các điều kiện của 1 cuộc tấn công CSRF, kẻ tấn công có thể lợi dụng lỗ hổng này để thực hiện mua hàng mà không mất tiền, ảnh hưởng tới tài chính của cửa hàng | Nghiêm trọng |
| 7 | Lỗ hổng IDOR xoá comment, xoá sản phẩm trong “giỏ hàng”, xoá đơn hàng ở “đơn của tôi” | Khi xoá comment, xoá sản phẩm trong “giỏ hàng”, xoá đơn hàng ở “đơn của tôi”, các request hiện rõ thuộc tính commentID, productID, kẻ tấn công có thể lợi dụng lỗ hổng này để xoá các bình luận, các đơn hàng khác của người dùng khác, làm giảm trải nghiệm người dùng, ảnh hưởng đến uy tín của cửa hàng | Cao |

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

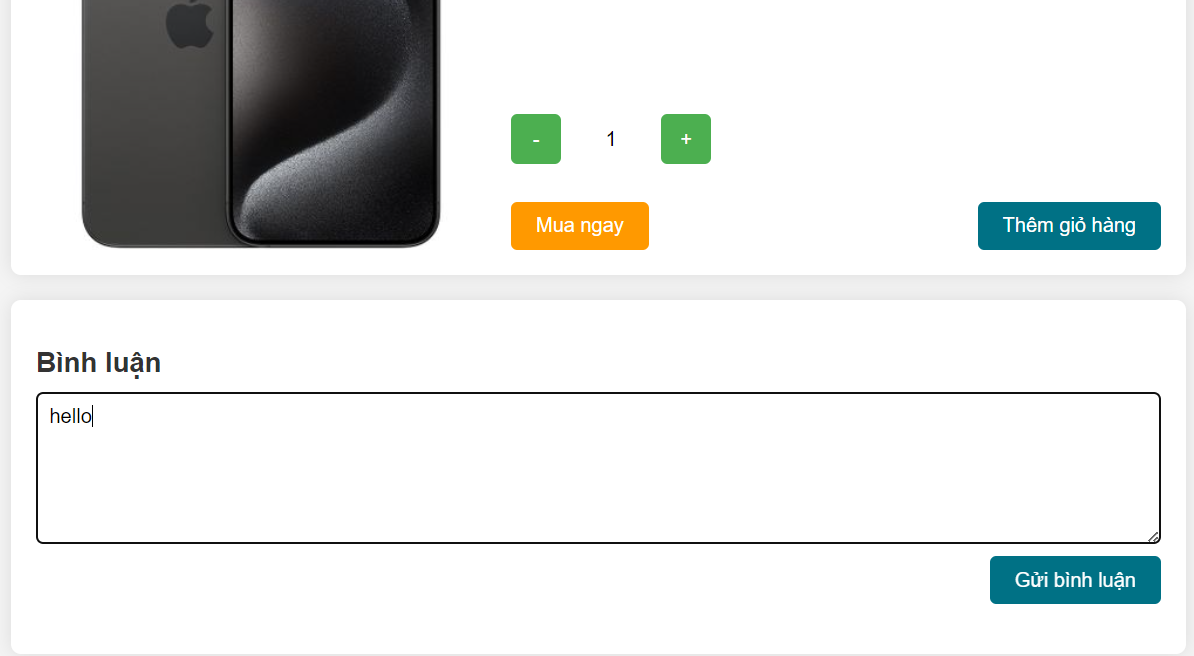
* **Mức độ: Cao**
* **Mô tả:** Đăng ký 1 tài khoản username có dạng <script>…</script>, khi comment sản phẩm sẽ bị lỗi XSS

Đăng ký tài khoản



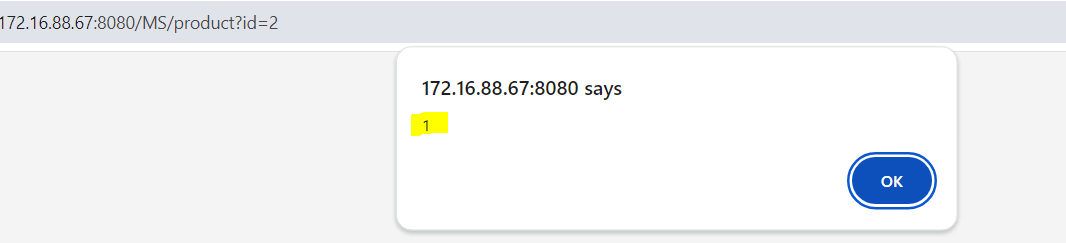
Hình 1: Đăng ký tài khoản với full-name bất thường

Sử dụng tài khoản vừa đăng ký, bình luận 1 sản phẩm bất kỳ



Hình 2: Dùng tài khoản vừa đăng ký bình luận

Bị lỗi XSS do hiện tên người dùng: bình luận, kẻ tấn công có thể lợi dụng để chiếm tài khoản người dùng



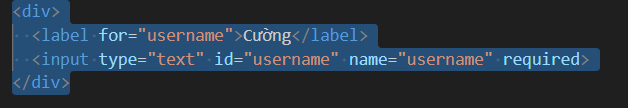
Hình 3: Trang web bị lỗi Stored-XSS

* **Khuyến nghị:**
* Kiểm tra đầu vào
* Không nên để full name ở phần comment mà để username hoặc gmail

1. **Lỗ hổng DOM-XSS trong cập nhật thông tin tài khoản**

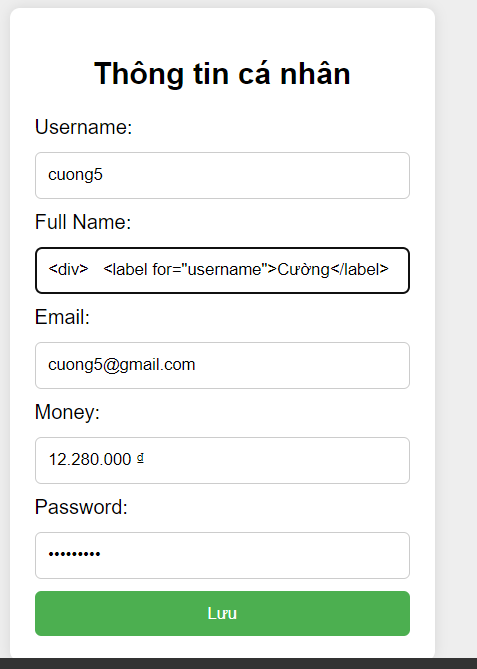
* **Mức độ: Cao**
* **Mô tả:** Cập nhật thông tin tài khoản ở các trường như fullname, … server không kiểm tra đầu vào, vì thế trang cập nhật thông tin tài khoản sẽ bị lỗi DOM-XSS

Code 1 form html bất kỳ



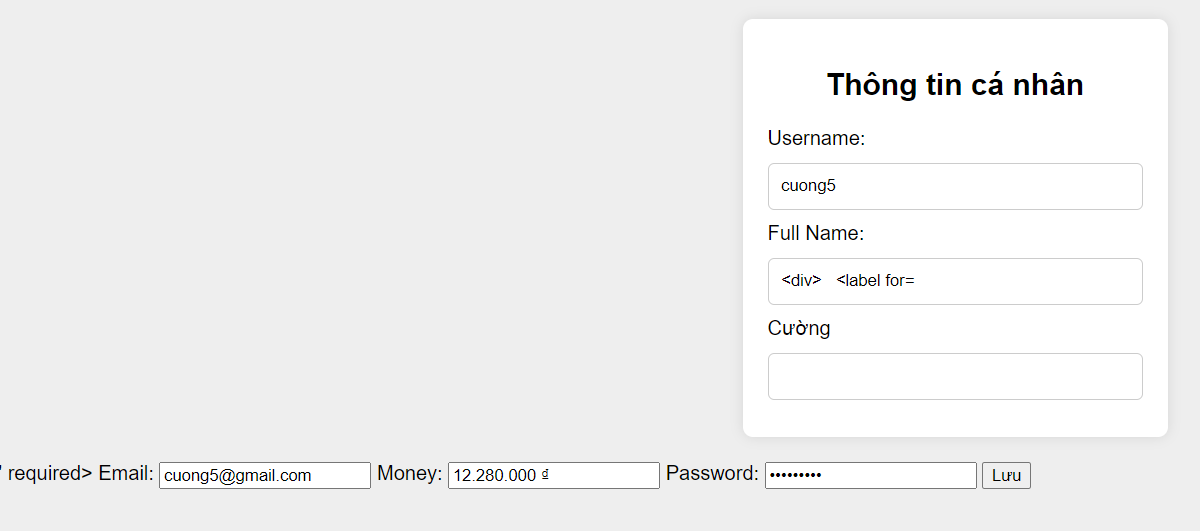
Hình 4. Code form html

Chỉnh sửa trường full-name trong trang cập nhật thông tin tài khoản



Hình 5. Dán vào phần bất kỳ có thể sửa trong “My Profile”

Khi ấn “Lưu” trang web bị lỗi DOM XSS



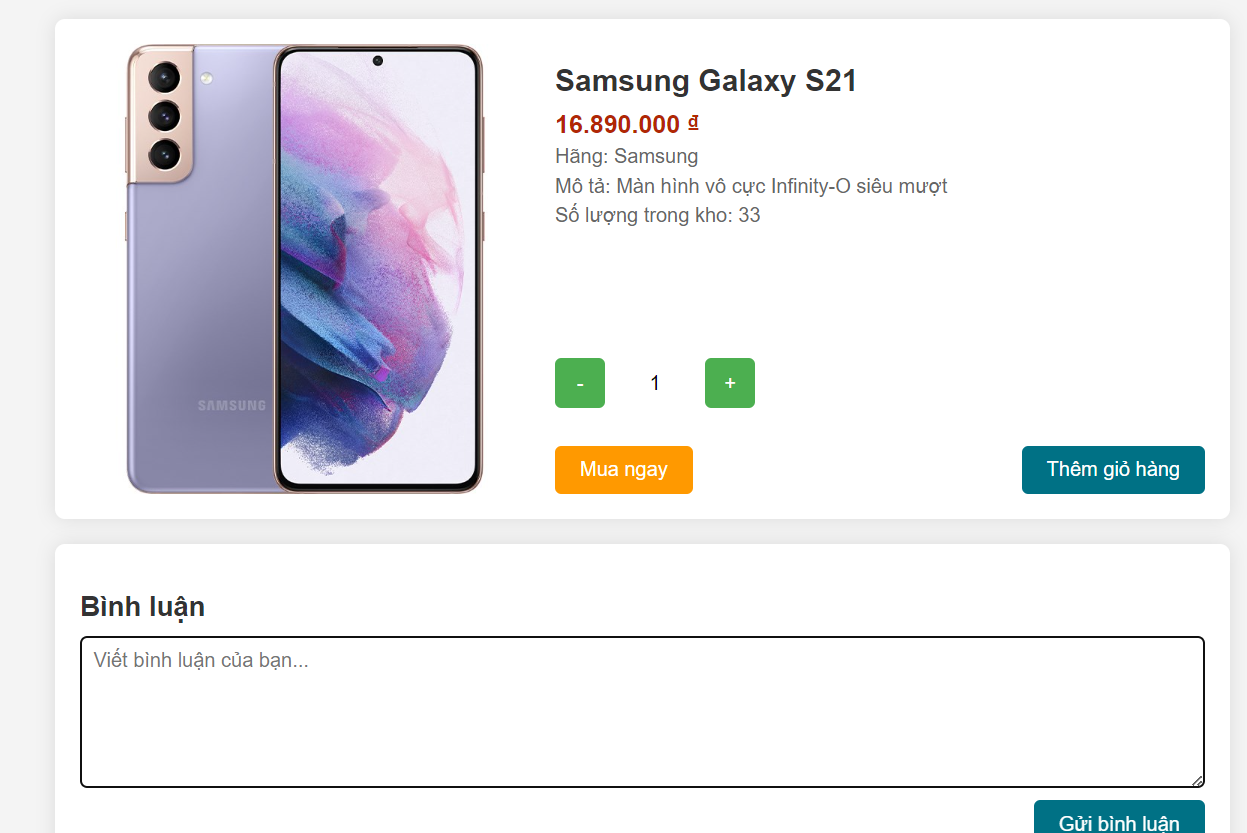
Hình 6. Trang web đã bị DOM-XSS

* **Khuyến nghị:**
  + Kiểm tra và làm sạch dữ liệu nhập từ người dùng.
  + Áp dụng CSP để hạn chế nội dung được tải và thực thi.
  + Thoát (escape) dữ liệu trước khi chèn vào DOM.
  + Sử dụng các framework hiện đại như React, Angular, Vue.js có cơ chế bảo vệ chống XSS.

1. **Lỗ hổng business logic có thể chỉnh sửa số lượng của sản phẩm trực tiếp trên trang web**

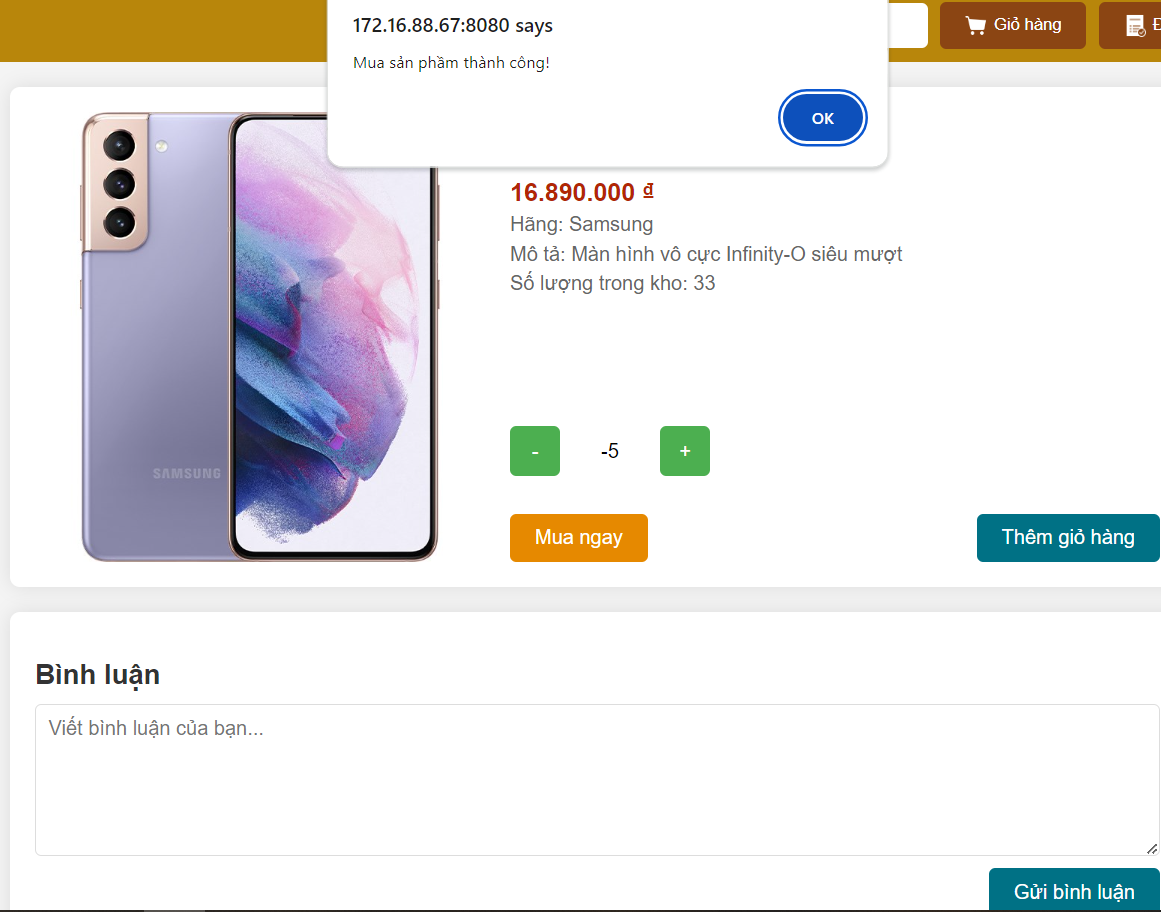
* **Mức độ: Nghiêm trọng**
* **Mô tả:** Việc không kiểm tra kỹ lưỡng đầu vào dẫn đến việc người dùng có thể chỉnh sửa số lượng sản phẩm thành số lượng âm, kéo theo giá tiền của sản phẩm cũng là số âm, người dùng không cần số dư tài khoản cũng có thể mua hàng 1 cách đơn giản.

Trang giao diện sản phẩm



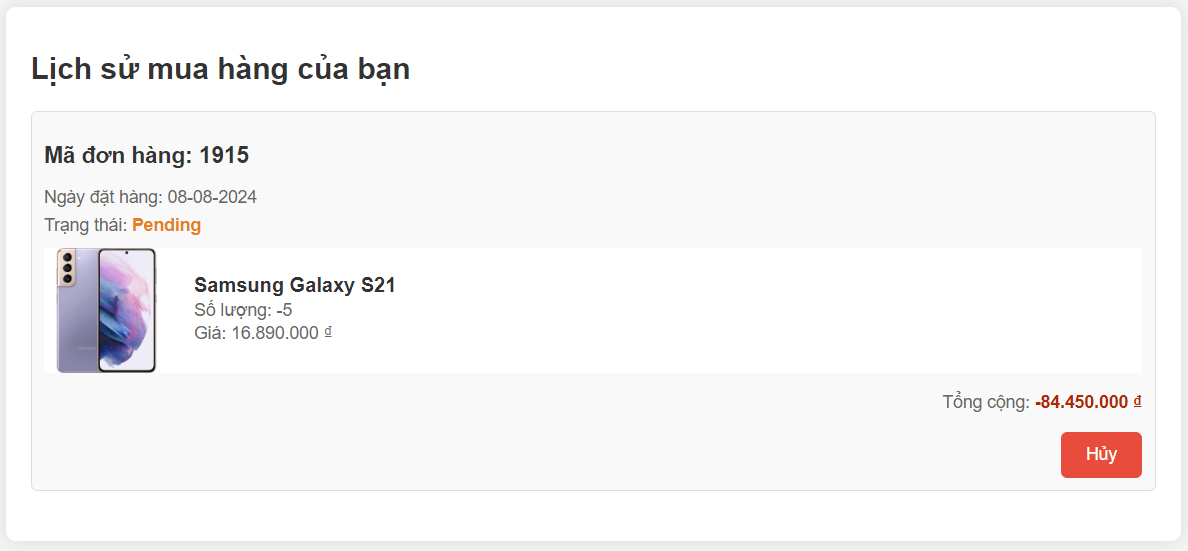
*Hình 7. Trang giao diện của sản phẩm SS S21*

Người dùng có thể chỉnh sửa số lượng về số âm để mua hàng và tiền trong tài khoản cũng tăng lê



Hình 8. Chỉnh số lượng thành “-5” và mua hàng thành công

Mua hàng thành công



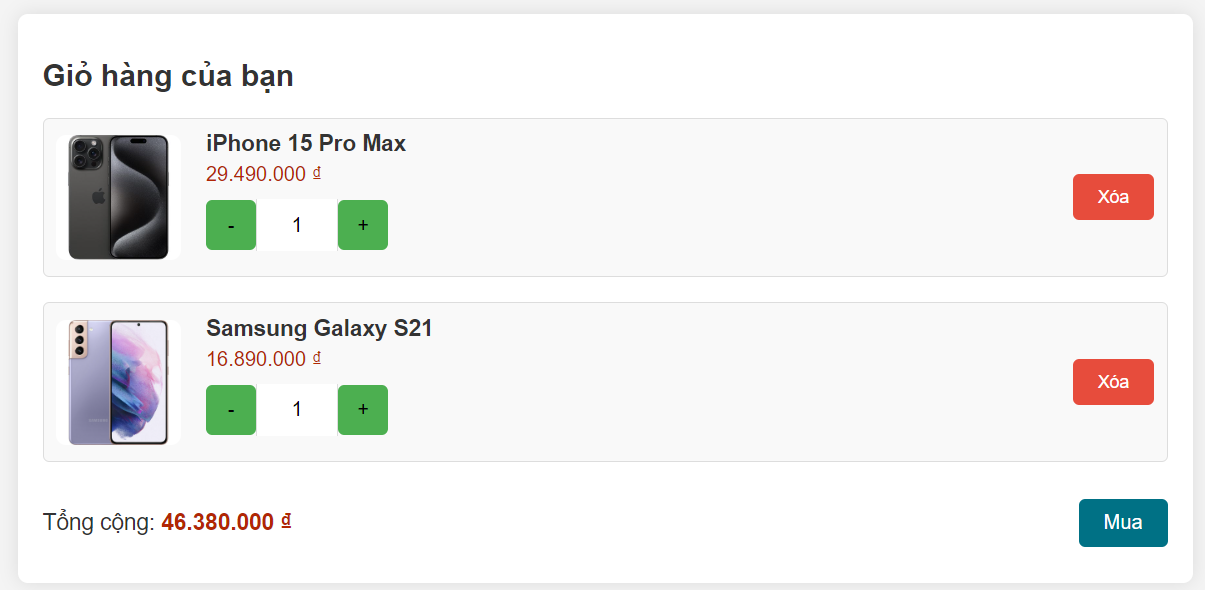
Hình 9. Lịch sử đơn hàng vừa mua thành công

* **Khuyến nghị:**
  + Kiểm tra và xác nhận rằng số lượng sản phẩm nhập vào là một số dương trước khi tiến hành mua hàng.
  + Giới hạn trường nhập liệu chỉ chấp nhận số nguyên dương. Sử dụng thuộc tính min trong các input số.
  + Thêm xử lý lỗi để thông báo cho người dùng khi nhập số lượng không hợp lệ.

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

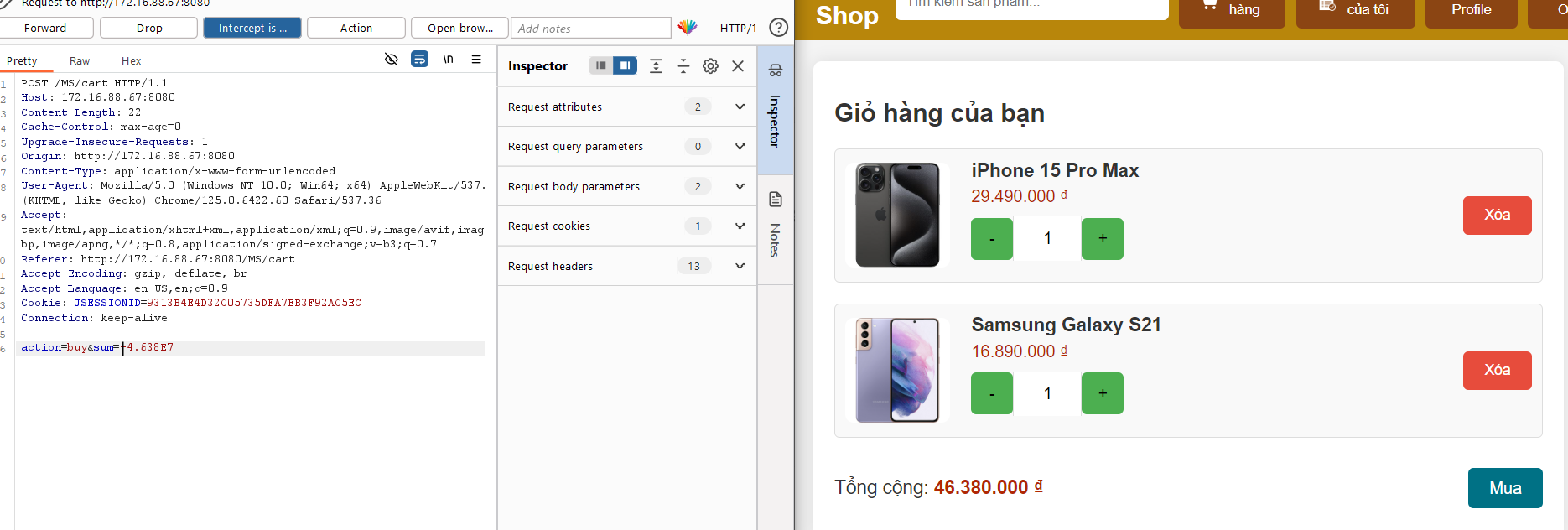
* **Mức độ: Nghiêm trọng**
* **Mô tả:** Bắt sự kiện mua hàng trong giỏ hàng, sửa giá tiền thành âm có thể mua hàng và số tiền trong tài khoản tự động tăng.

Thêm nhiều hơn 1 sản phẩm bất kỳ vào giỏ hàng:



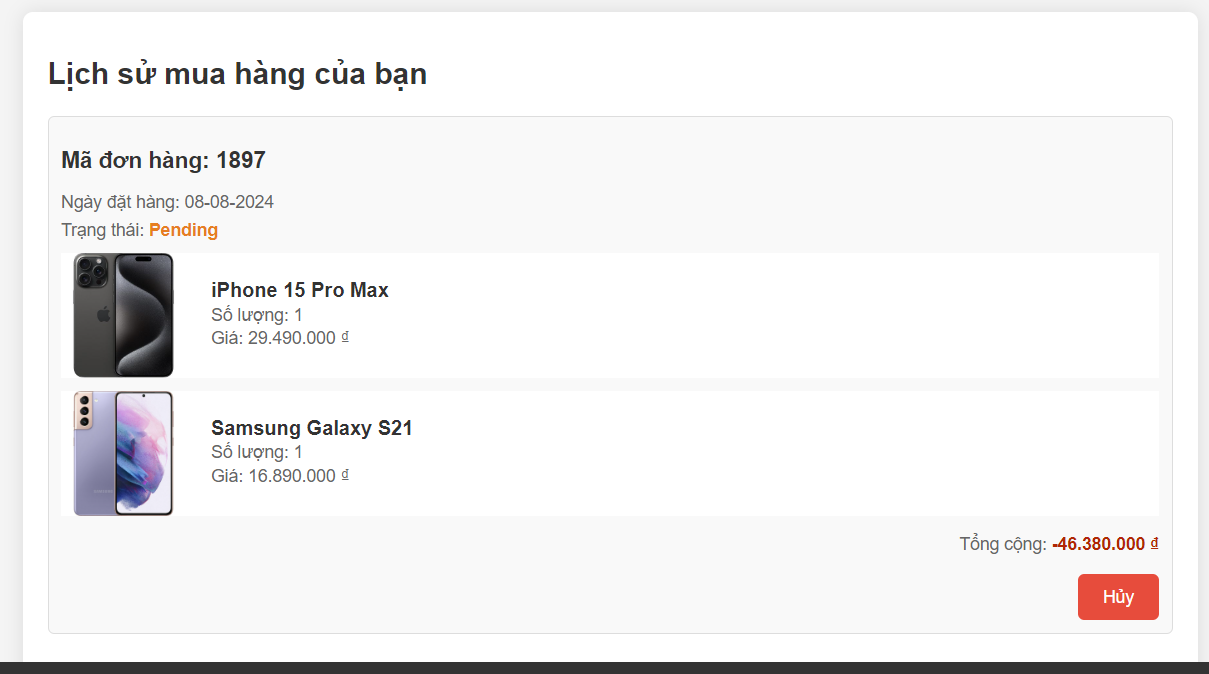
Hình 10.Giỏ hàng

Bật intercept, bắt sự kiện nút “Mua” sửa thành tiền âm, sau đó ấn Forward



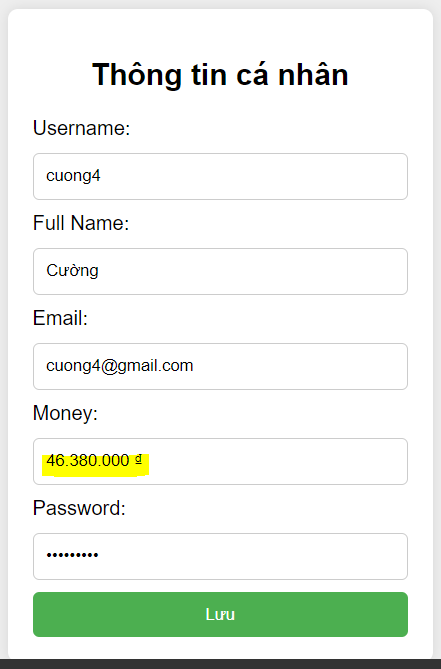
*Hình 11. Sửa giá tiền từ 4.63E7 thành -4.63E7*

Mua hàng thành công với tiền âm



*Hình 12. Mua hàng thành công*

Trong “My Profile” số tiền từ 0 đã được cộng



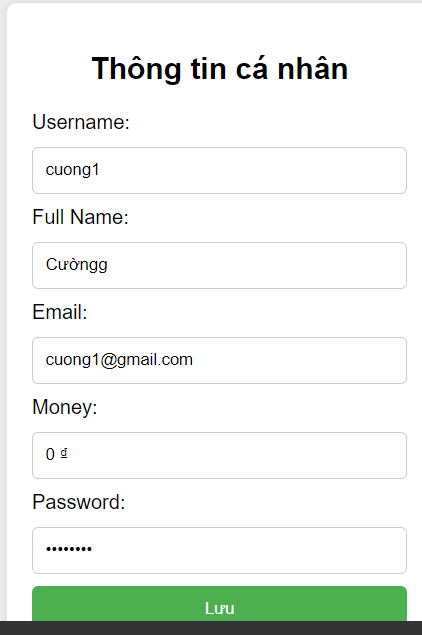
Hình 13. Số tiền đã được cập nhật

* **Khuyến nghị:**
  + Kiểm tra và xác nhận rằng giá trị tiền phải là số dương trước khi tiến hành mua hàng.
  + Tránh cho phép người dùng chỉnh sửa trực tiếp giá trị tiền; thay vào đó, giá trị tiền nên được xác định từ cơ sở dữ liệu và chỉ đọc.
  + Xoá trường “sum” và thêm trạng thái “chờ duyệt”.

1. **Lỗ hổng CSRF trong cập nhật thông tin cá nhân**

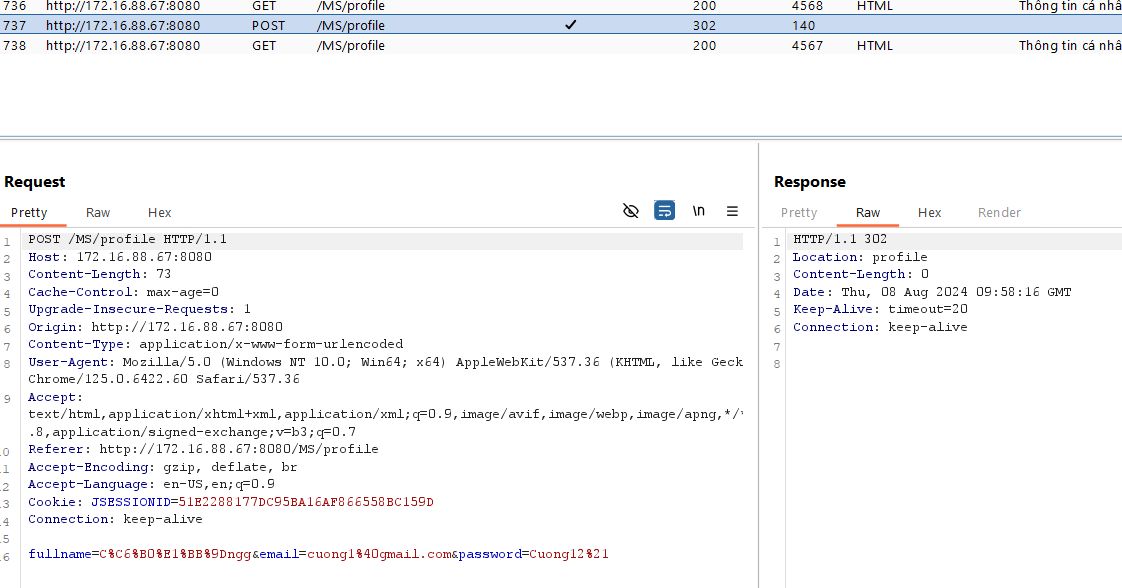
* **Mức độ: Cao**
* **Mô tả:** Request POST /MS/profile có thể lợi dụng để kẻ tấn công thực hiện 1 cuộc tấn công CSRF, khi người dùng không may truy cập link độc hại này, kẻ tấn công có thể đánh cắp tài khoản người dùng

Kẻ tấn công thực hiện sửa thông tin cá nhân



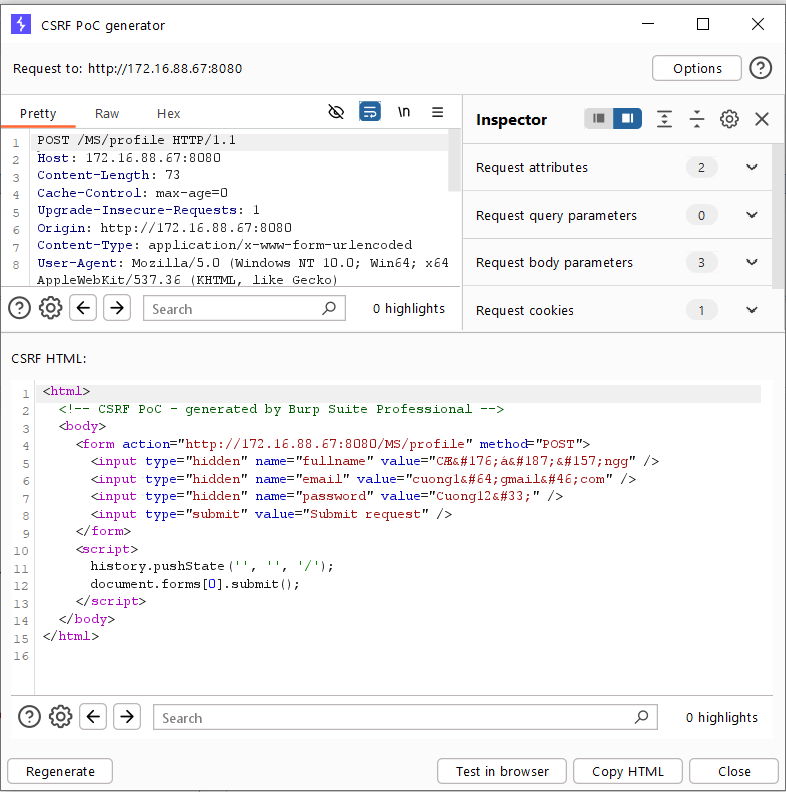
Hình 14. Sửa thông tin của cuong1

Bắt request cập nhật thông tin cá nhân



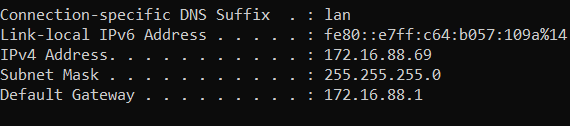
Hình 15. Request sửa thông tin cá nhân của cuong1

Tạo yêu cầu độc hại



Hình 16. Tạo yêu cầu độc hại

IP của máy tính kẻ tấn công:



Hình 17. IP là 172.16.88.69

Chạy code đặt tên file code bất kỳ (vi.html) và gửi link: <http://172.16.88.69:5500/vi.html> cho nạn nhân,

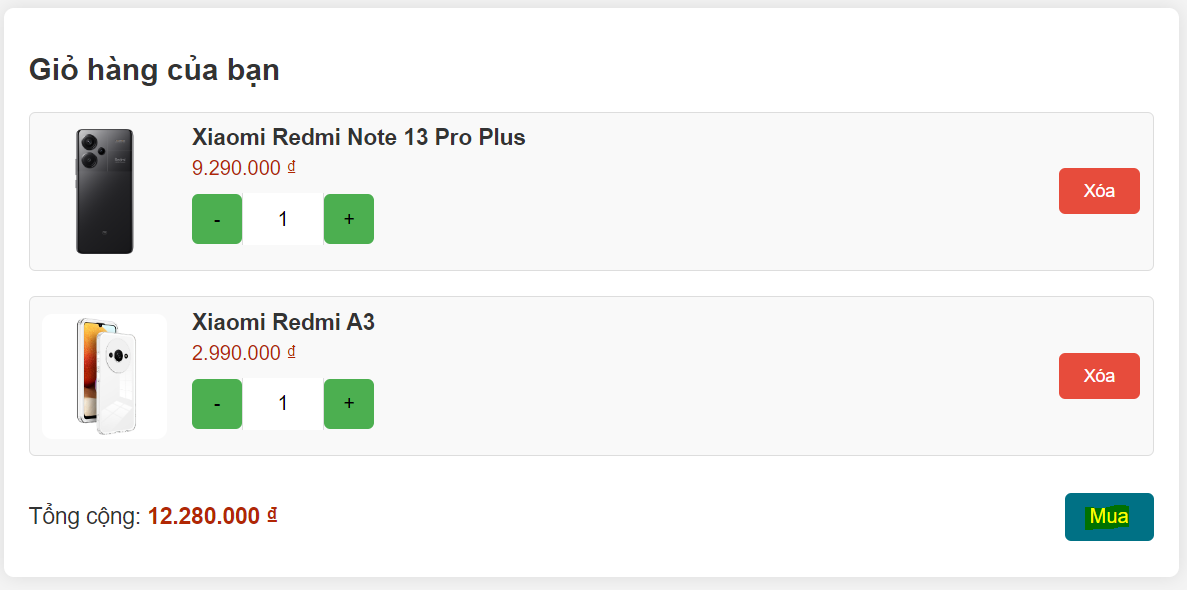
|  |  |
| --- | --- |
| Hình 18. Profile khi chưa ấn yêu cầu độc hại | Hình 19. Profile sau khi click yêu cầu độc hại |

* **Khuyến nghị:**
  + Sử dụng CSRF Token.
  + Sử dụng SameSite Attribute cho Cookies.
  + Yêu cầu xác thực lại
  + Sử dụng 2FA, captcha.

1. **Lỗ hổng CSRF trong mua hàng ở giỏ hàng**

* **Mức độ: Nghiêm trọng**
* **Mô tả:** Request POST MS/cart thoả mãn các điều kiện của 1 cuộc tấn công CSRF, kẻ tấn công có thể lợi dụng lỗ hổng này để thực hiện mua hàng mà không mất tiền, ảnh hưởng tới tài chính của cửa hàng

Mua thử sản phẩm trong giỏ hàng



Hình 20. Thêm sản phẩm vào giỏ hàng

Bắt request



Hình 21. Bắt request thêm vào giỏ hàng

Gửi sang Repeater để phân tích

Thoả mãn 3 điều kiện của 1 cuộc tấn công CSRF



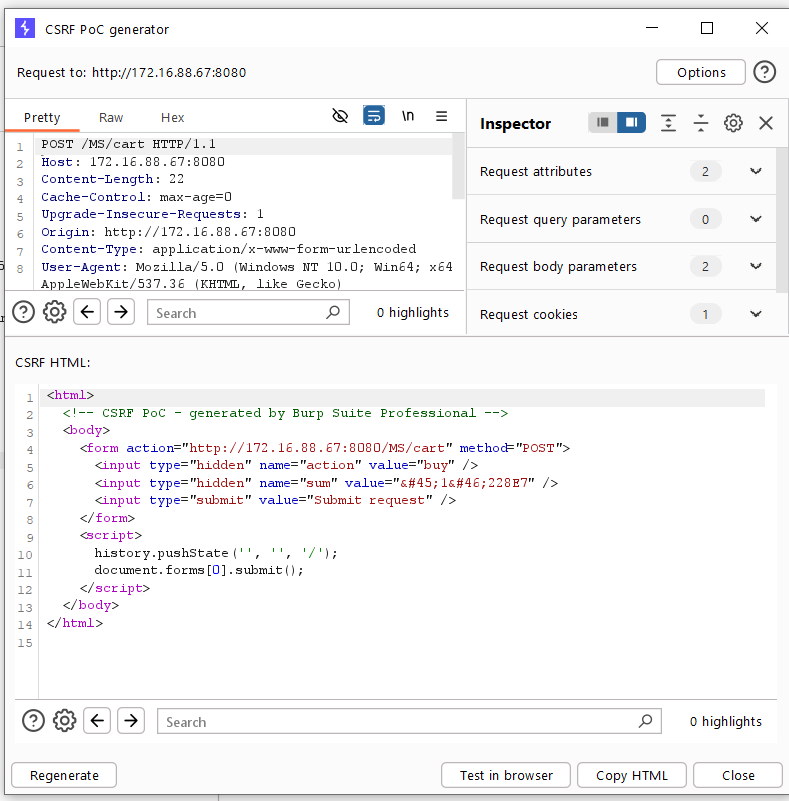
Hình 22. Request thêm vào giỏ hàng

Sửa giá tiền thành âm, lỗi này sẽ được sử dụng như 1 tool tự động tăng tiền cho người dùng

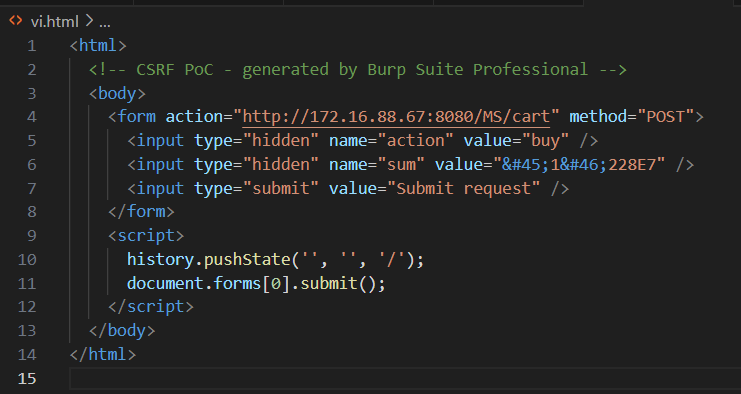


Hình 23. Sửa thành giá tiền âm

Tạo mã CSRF

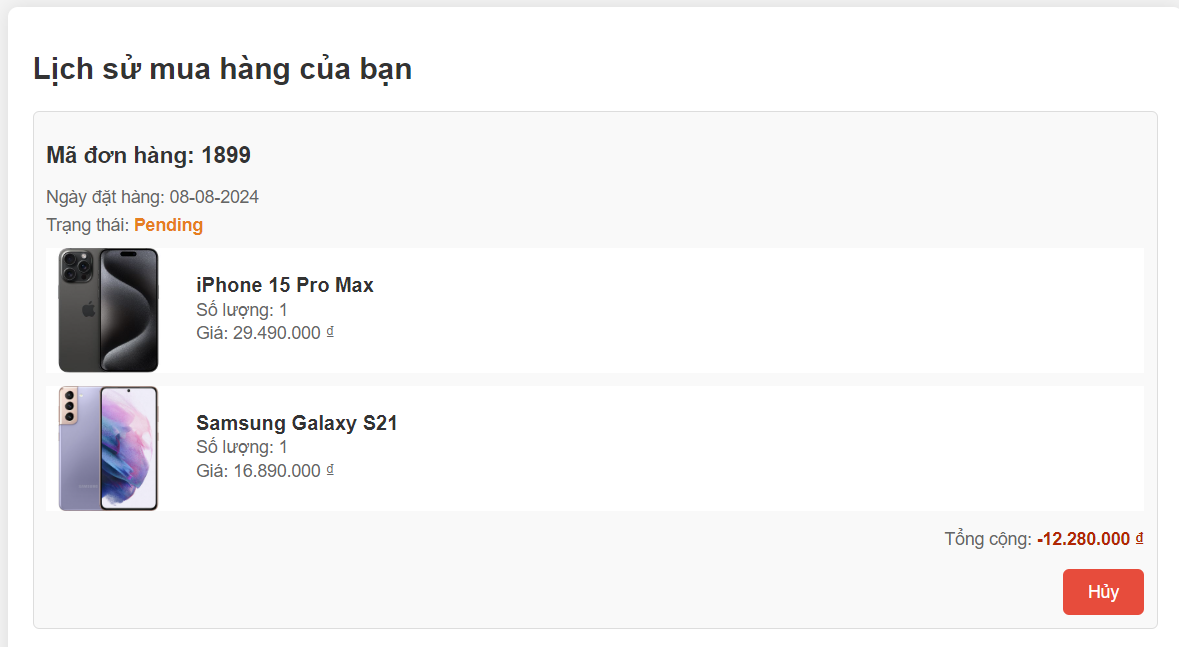


Hình 24. Tạo CSRF PoC

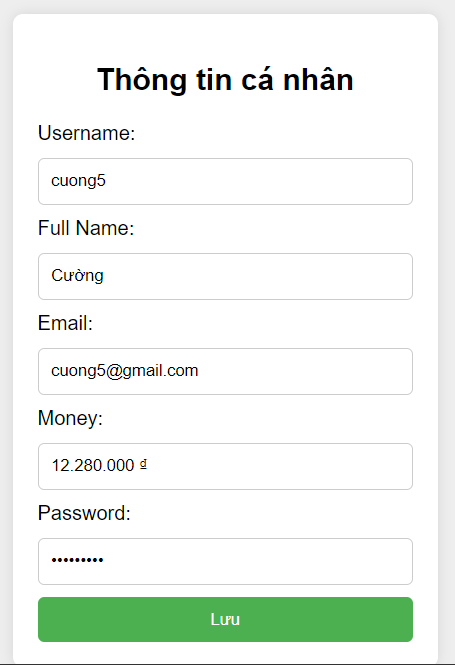


Hình 25. Chạy code html

Gửi link <http://172.16.88.69:5500/vi.html> cho người dùng, người dùng thêm các sản phẩm tương tự như trên vào giỏ hàng, người dùng click vào link sẽ tự động mua hàng



Hình 26. Tự động mua hàng với giá tiền âm



Hình 27. Số tiền trong tài khoản được cập nhật

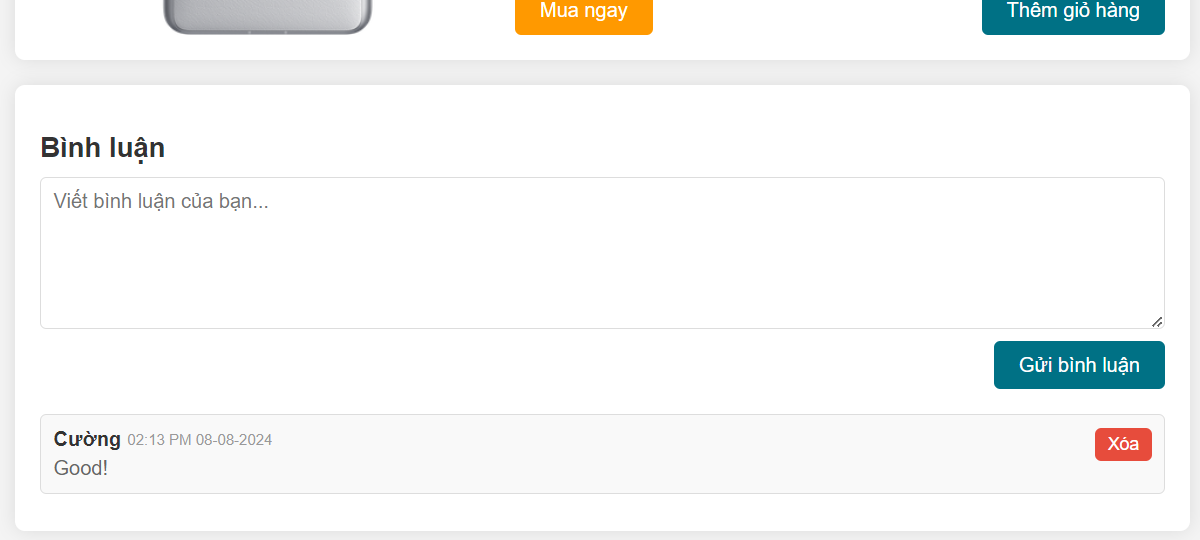
Lỗi CSRF còn có ở tự động comment, …

* **Khuyến nghị:**
  + Sử dụng CSRF Token.
  + Xác thực người dùng trước khi thực hiện giao dịch.
  + Sử dụng captcha cho giao dịch.

1. **Lỗ hổng IDOR xoá comment, xoá sản phẩm trong “giỏ hàng”, xoá đơn hàng ở “đơn của tôi”.**

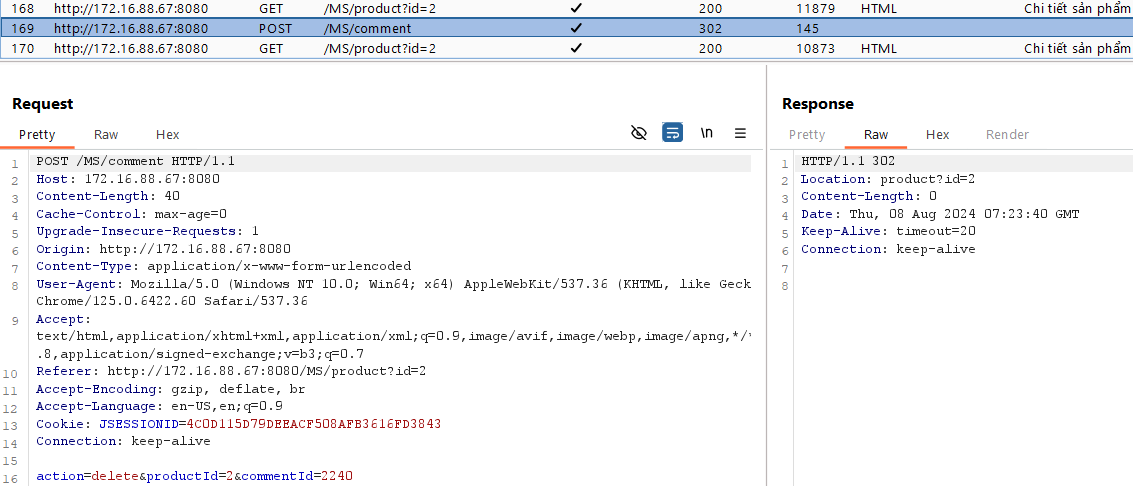
* **Mức độ: Cao**
* **Mô tả:** Khi xoá comment, xoá sản phẩm trong “giỏ hàng”, xoá đơn hàng ở “đơn của tôi”, các request hiện rõ thuộc tính commentID, productID, kẻ tấn công có thể lợi dụng lỗ hổng này để xoá các bình luận, các đơn hàng khác của người dùng khác, làm giảm trải nghiệm người dùng, ảnh hưởng đến uy tín của cửa hàng

Bình luận 1 comment bất kỳ và xoá comment đó



Hình 28. Bình luận bất kỳ, sau đó xoá bình luận

Bắt request:

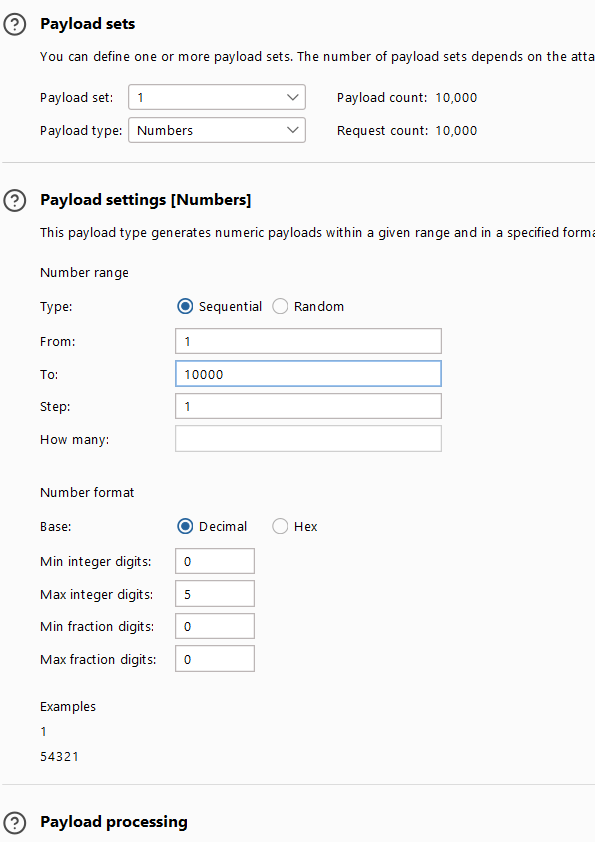


Hình 29. Request xoá comment

Gửi sang Intruder để brute-force, xoá tất cả comment



Hình 30. Brute-force commentID



Hình 31. Brute-force id từ 1 tới 10000

Start Attack -> Tất cả các comment đã bị xoá

Tương tự với Xoá sản phẩm trong “Giỏ hàng” và xoá đơn hàng ở “Đơn của tôi”

* **Khuyến nghị:** 
  + Mã hoá các trường id.
  + Sử dụng kiểm tra dựa trên phiên người dùng
  + Thực hiện rate limiting để ngăn chặn các yêu cầu ddos.