BÁO CÁO BÀI TẬP LÝ THUYẾT

**Môn học: Bảo mật web và ứng dụng**

**Kỳ báo cáo: Buổi 03 (Session 03)**

**Tên chủ đề: Path traversal – OS command injection**

*GVHD: Nghi Hoàng Khoa*

*Ngày báo cáo: 04/04/2024*

**Nhóm: 03 (ghi số thứ tự nhóm)**

1. **THÔNG TIN CHUNG:**

*(Liệt kê tất cả các thành viên trong nhóm)*

Lớp: NT213.O21.ANTN

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Phạm Ngọc Thơ | 21522641 | 21522641@gm.uit.edu.vn |
| 2 | Hà Thị Thu Hiền | 21522056 | 21522056@gm.uit.edu.vn |

1. **NỘI DUNG THỰC HIỆN:[[1]](#footnote-1)**

|  |  |  |
| --- | --- | --- |
| **STT** | **Công việc** | **Kết quả tự đánh giá** |
| 1 | Path traversal (quay video) | 100% |
| 2 | OS command injection (trình bày trong file pdf) | 100% |

**Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.**

BÁO CÁO CHI TIẾT

Truy cập lab tại: <https://portswigger.net/web-security/all-labs>

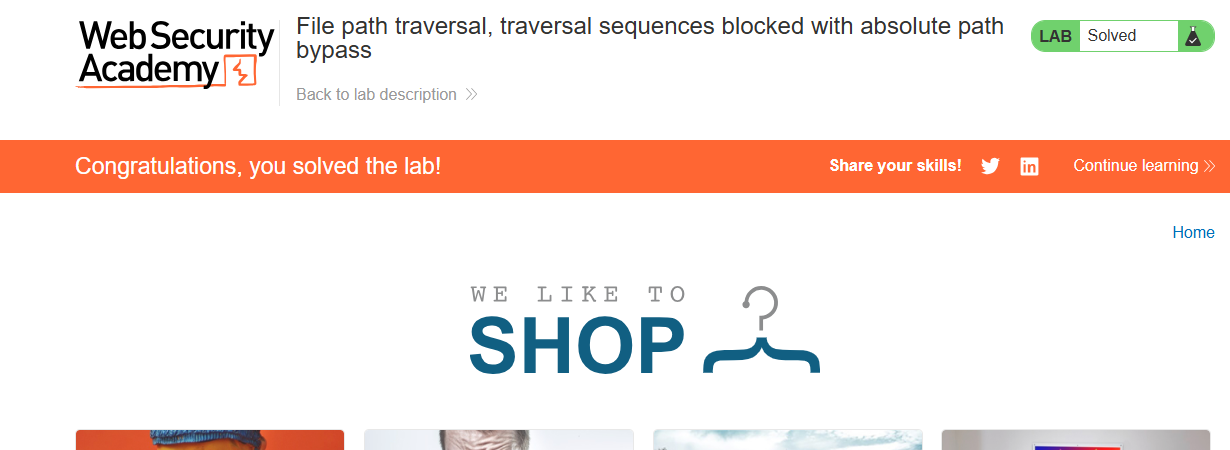
1. **Path traversal:**
2. **File path traversal, simple case:**

* **Link:** [**https://youtu.be/bE89bWddmiE**](https://youtu.be/bE89bWddmiE)



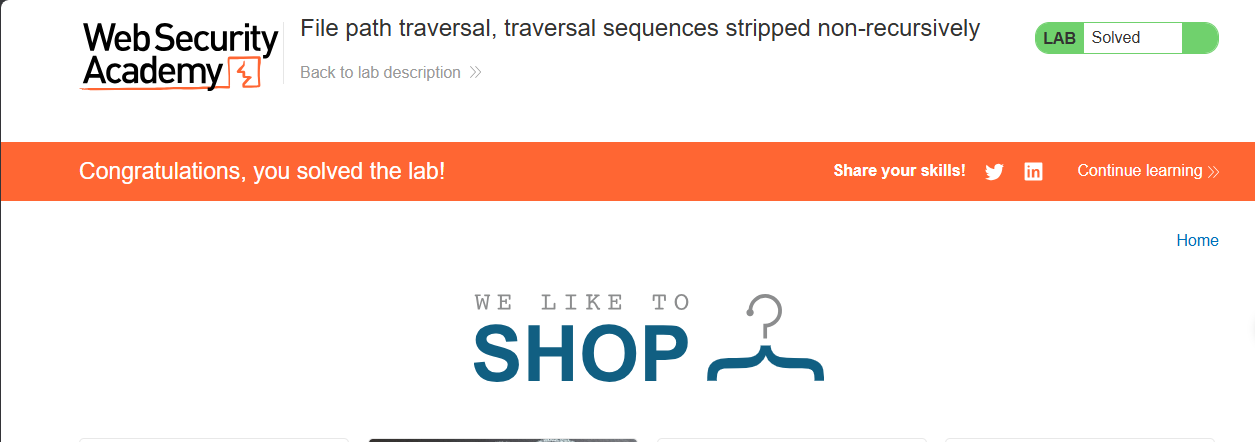
1. **File path traversal, traversal sequences blocked with absolute path bypass:**

* **Link:** [**https://youtu.be/Juns\_S2T0E0**](https://youtu.be/Juns_S2T0E0)



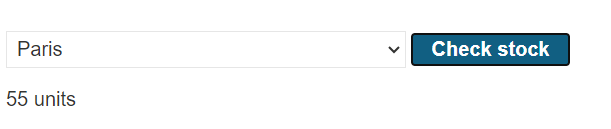
1. **File path traversal, traversal sequences stripped non-recursively:**

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

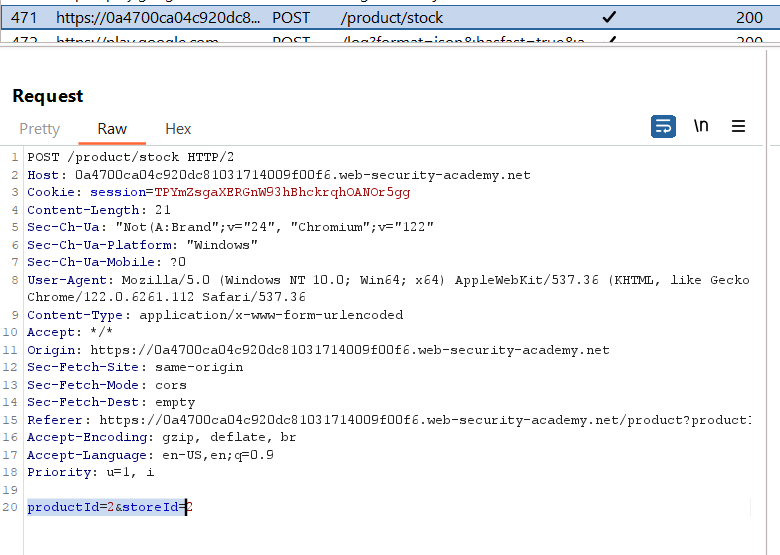


1. **OS command injection:**
2. **OS command injection, simple case**

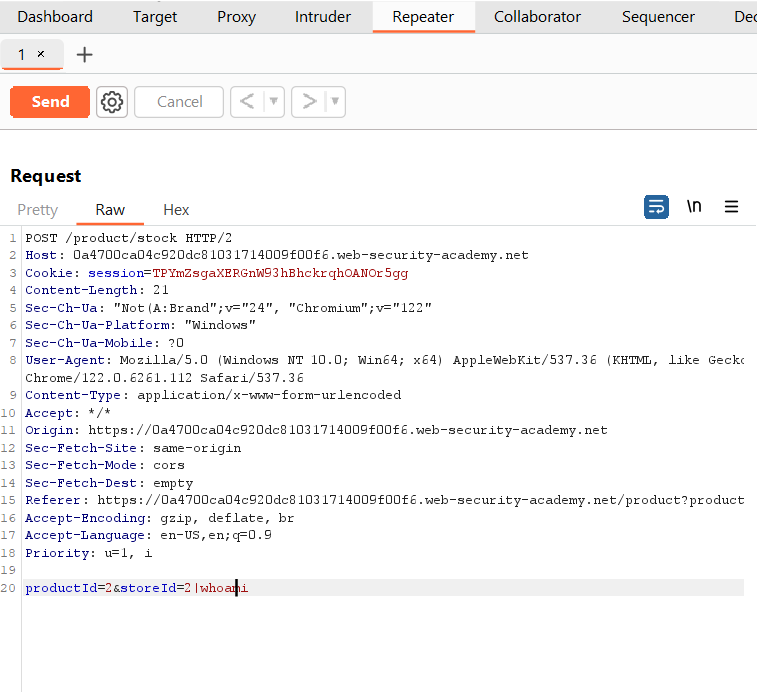
* Đề bài: Lab này chứa lỗ hổng chèn lệnh hệ điều hành trong trình kiểm tra kho sản phẩm. Ứng dụng thực thi lệnh shell hiển thị ID sản phẩm và ID cửa hàng, đồng thời trả về kết quả thô từ lệnh trong response. Để giải bài lab, hãy thực thi lệnh *whoami* để xác định tên của người dùng hiện tại.
* Giao diện ban đầu sẽ như thế này:



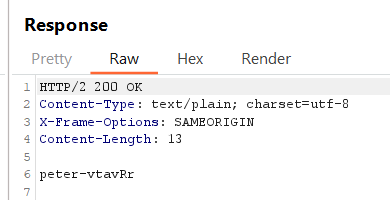
* Em sử dụng Burp Suite Pro để xem các gói tin request và response với trang web. Khi kiểm tra stock, thì trong gói tin request có tiết lộ ID sản phẩm và ID cửa hàng:



* Để sửa request, em sẽ sử dụng Repeater. Tiến hành chèn thêm lệnh *whoami* như bài lab yêu cầu, để không bị dính với storedId làm hỏng gói tin, em sẽ nối lệnh cần chèn vào phía sau ID:



* Kết quả: tìm được “peter-vtarRr”





1. **Blind OS command injection with time delays**

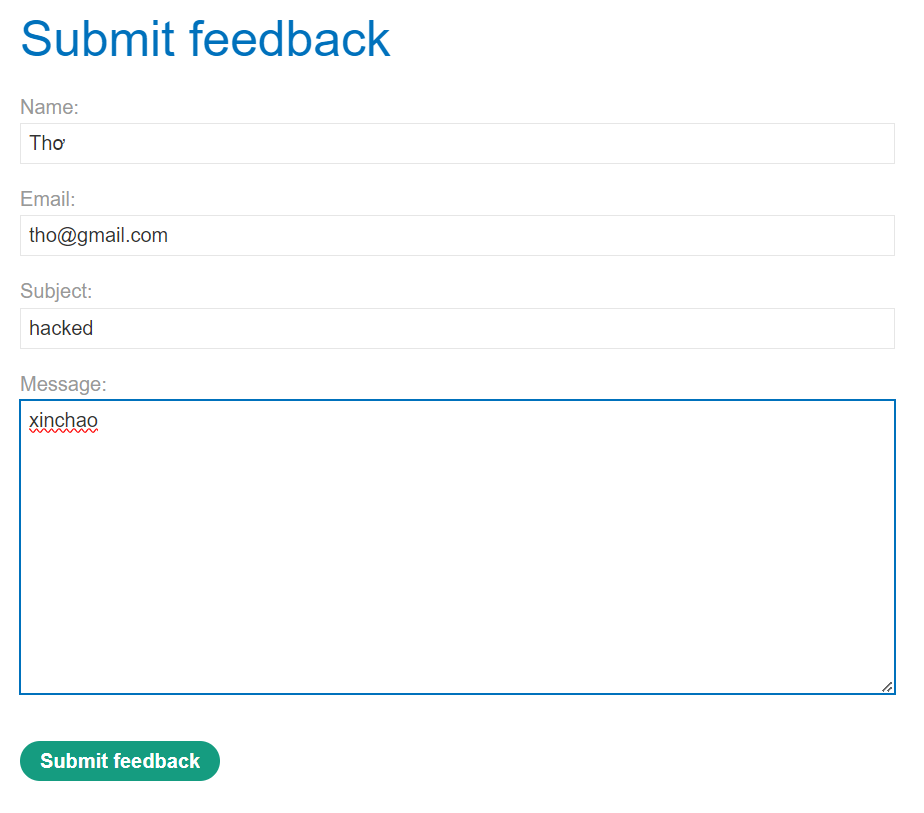
* Đề bài: Lab này chứa lỗ hổng blind OS command injection trong chức năng feedback. Ứng dụng thực thi lệnh shell chứa các details. Đầu ra từ lệnh không được trả về trong response. Để giải quyết vấn đề trong lab này, hãy khai thác lỗ hổng blind OS command injection để gây ra độ trễ 10 giây.
* Blind OS command injection: là lỗ hổng mà đầu ra không trả về trong response và ouput sẽ không hiển thị trên màn hình (hay còn gọi là 1 lỗ hổng tàng hình). Có vài kỹ thuật được dùng có thể sử dụng để xác định được blind OS command, trong đó phải kể đến là kỹ thuật được dùng trong bài này: Sử dụng time delays để xác định được blind vulnerabilities, cho phép chúng ta xác nhận rằng lệnh này đã được thực thi hay chưa dựa vào thời gian mà ứng dụng cần để đáp ứng.

Tham khảo tại <https://viblo.asia/p/os-command-injection-thuc-thi-shell-code-cung-idol-L4x5xrXbZBM> lệnh thường được sử dụng để tạo delays là:

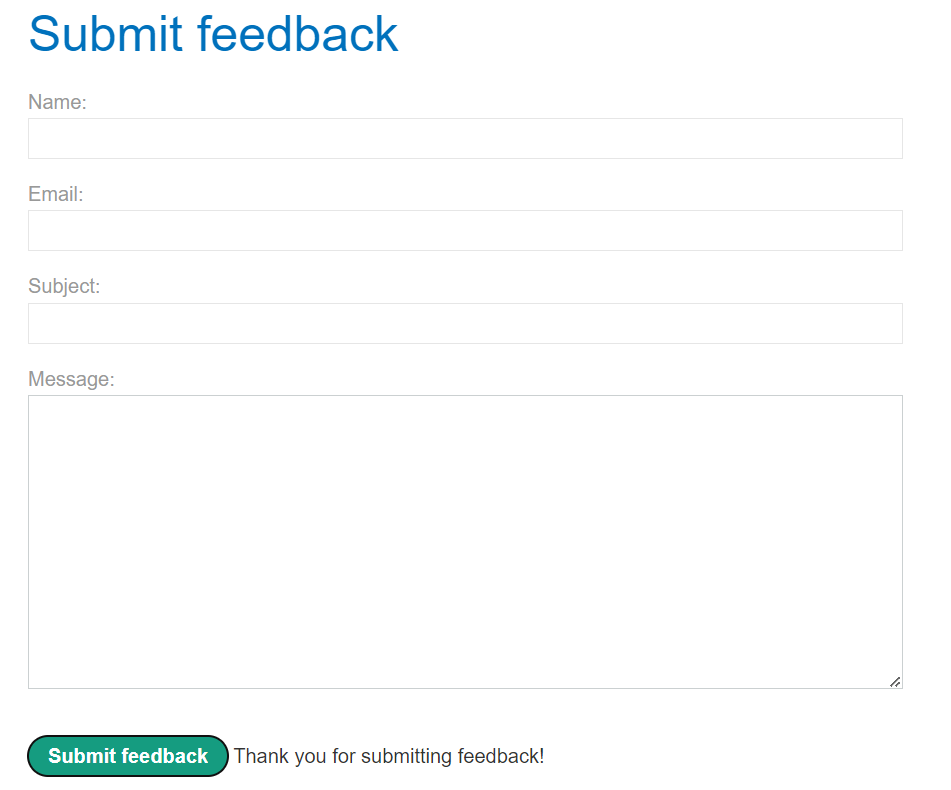
ping -c 10 127.0.0.1

Và em cũng sẽ áp dụng kỹ thuật này cho lab này.

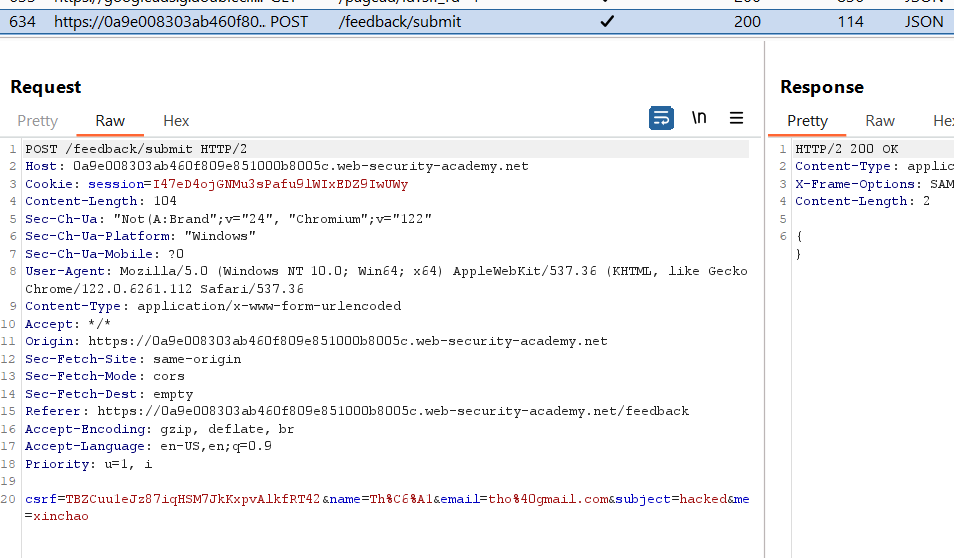
* Đầu tiên, mở giao diện feedback thông qua browser của Burp:



* Trang phản hồi:



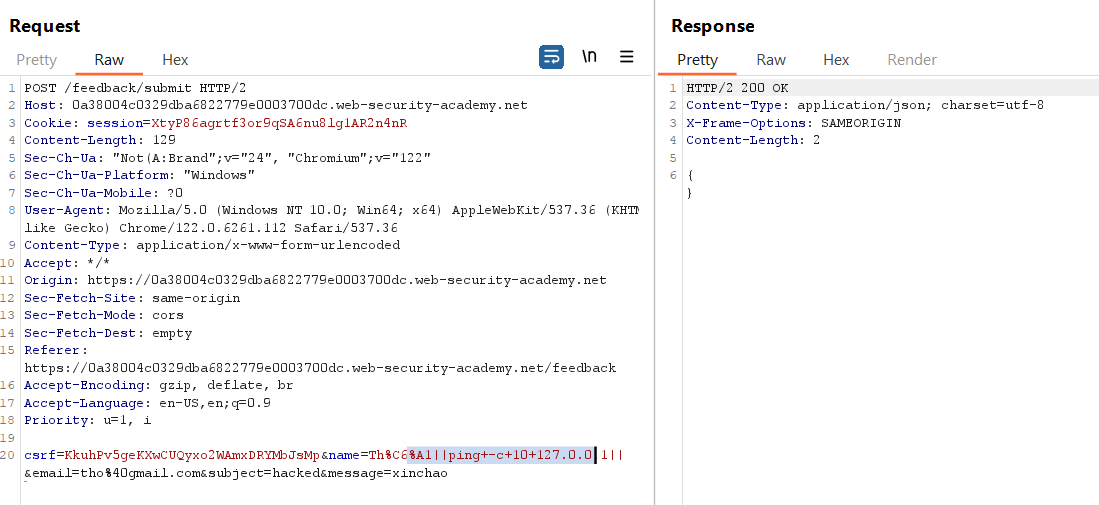
* Kiểm tra HTTP History, thấy đây chính là gói tin để gửi feedback:



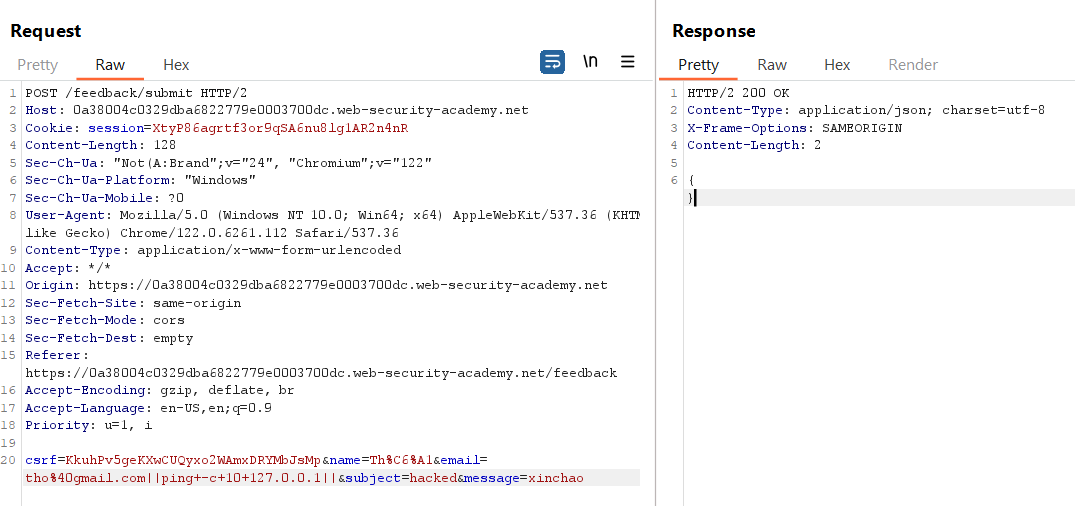
* Có bốn trường thông tin: name, email, subject và message - đều do người dùng nhập vào. Tuy nhiên, để xác định được trường nào có thể khai thác, em sẽ đặt lệnh trên lần lượt vào các trường này. Câu lệnh cần chèn:

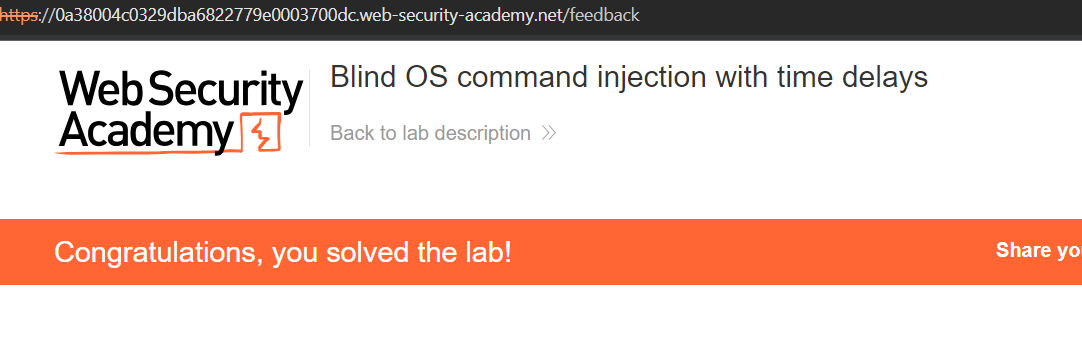
||ping+-c+10+127.0.0.1||

Trong URL encoding, space sẽ được thay bằng ký tự +. Và ký tự || dùng để chèn một phần của lệnh SQL tiếp theo. Em sẽ đặt sau trường *name*, kết quả trả về là một cặp ngoặc nhọn:



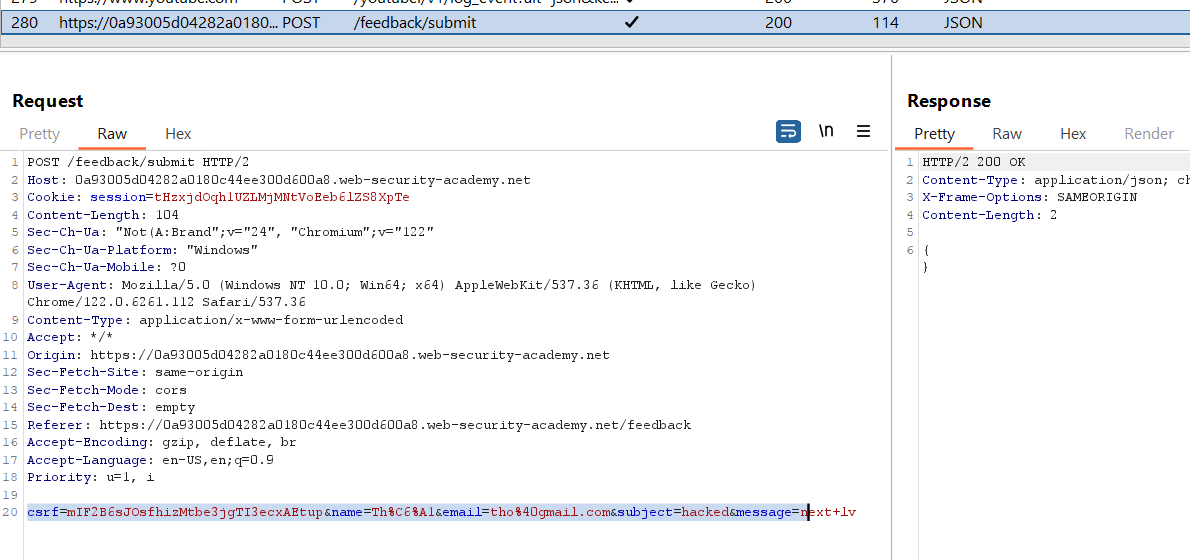
* Đặt sau trường *email****,*** chương trình sẽ không phản hồi ngay lập tức mà dừng lại một chút, sau đó sẽ phản hồi tương tự như trên:





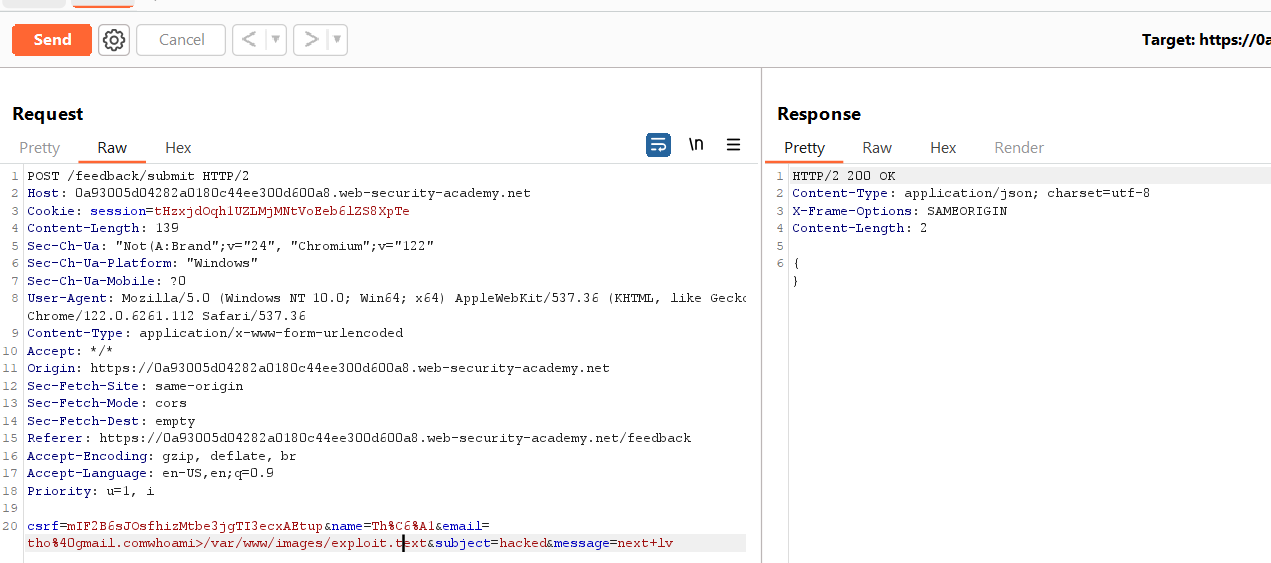
1. **Blind OS command injection with output redirection**

* Đề bài: Lab này chứa lỗ hổng blind OS command injection trong chức năng feedback. Ứng dụng thực thi lệnh shell chứa các thông tin do người dùng nhập. Đầu ra từ lệnh không được trả về trong phản hồi. Tuy nhiên, bạn có thể sử dụng chuyển hướng đầu ra để nắm bắt đầu ra từ lệnh. Có một thư mục có thể ghi tại: */var/www/images/*. Ứng dụng cung cấp hình ảnh cho danh mục sản phẩm từ vị trí này. Bạn có thể chuyển hướng đầu ra từ lệnh được chèn sang một tệp trong thư mục này, sau đó sử dụng URL tải hình ảnh để truy xuất nội dung của tệp. Để giải bài lab, hãy thực hiện lệnh *whoami* và truy xuất kết quả.
* Sau khi điền feedback thì trang cũng phản hồi tương tự như challenge phía trên.
* Gói tin request chứa thông tin em đã điền:

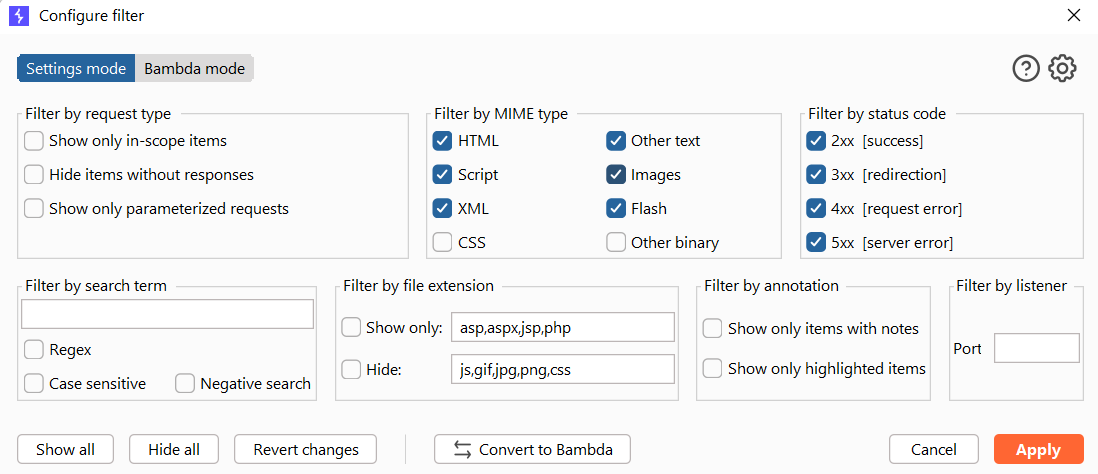


* Gửi qua Repeater và thực hiện thay đổi gói tin request. Lần này em sẽ thử tại vị trí email trước vì trong lab trước, đây là vị trí dễ bị tấn công. Theo như yêu cầu, em sẽ sử dụng lệnh *whoami* để tìm thông tin user và chuyển hướng lưu kết quả của lệnh vào 1 file trong thư mục */var/www/images/*:

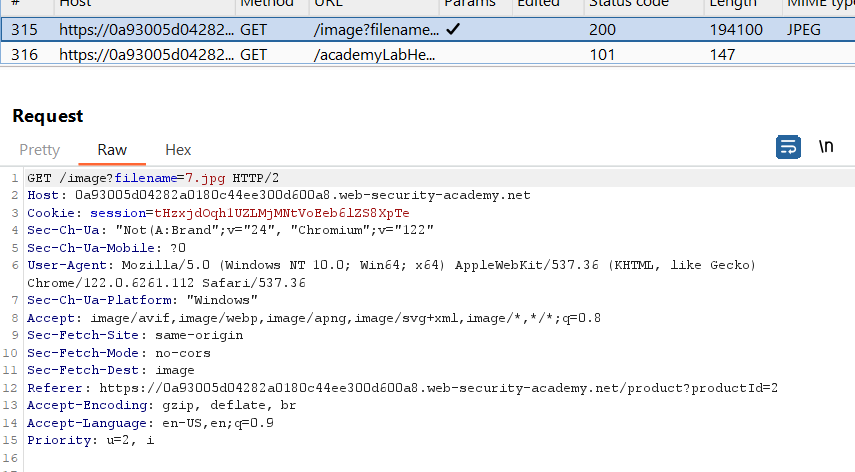
whoami>/var/www/images/exploit.text

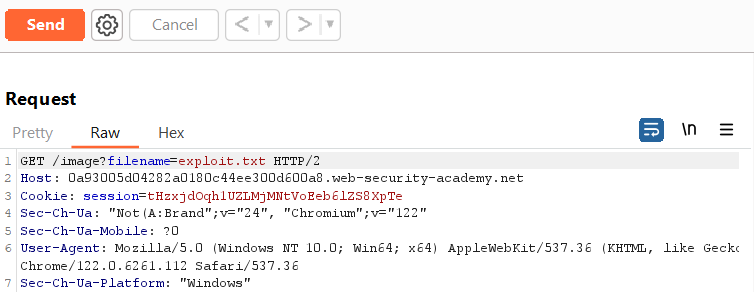


* Để xem được kết quả, cần phải mở được file *exploit.txt*, nhưng đường dẫn đầy đủ đến file thì ta lại không biết. Nhưng ta có thể tận dụng cách ứng dụng bao gồm các hình ảnh của nó với yêu cầu GET tới /image?filename=exploit.txt.
* Sử dụng filter để lọc image:

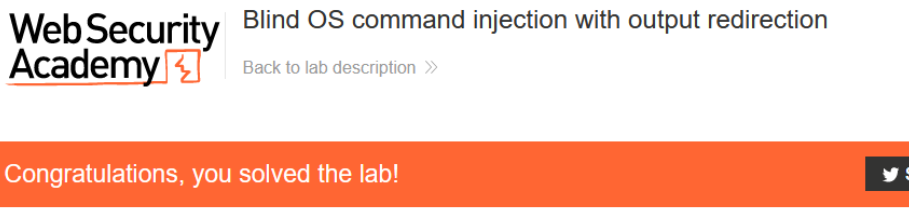


* Chọn được gói tin, tiến hành sửa đổi nội dung GET:





* Kết quả:



---

***Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này***

# **YÊU CẦU CHUNG**

* Sinh viên tìm hiểu và thực hành theo hướng dẫn.
* Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
* Sinh viên báo cáo kết quả thực hiện và nộp bài.

**Báo cáo:**

* File .DOCX và .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
* Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
* Đặt tên theo định dạng: [Mã lớp]-SessionX\_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành đã đăng ký với GVHD-TH).

*Ví dụ: [*NT101.K11.ATCL*]-Session1\_Group3.*

* Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
* Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm bài Lab.
* Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

**Đánh giá**: Sinh viên hiểu và tự thực hiện được bài thực hành. Khuyến khích:

* Chuẩn bị tốt.
* Có nội dung mở rộng, ứng dụng trong kịch bản phức tạp hơn, có đóng góp xây dựng bài thực hành.

*Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.*

**HẾT**

1. Ghi nội dung công việc, các kịch bản trong bài Thực hành [↑](#footnote-ref-1)