**Directory traversal**

Path traversal, also known as directory traversal, is a type of vulnerability that allows an attacker to read arbitrary files on the server running an application. This can include:

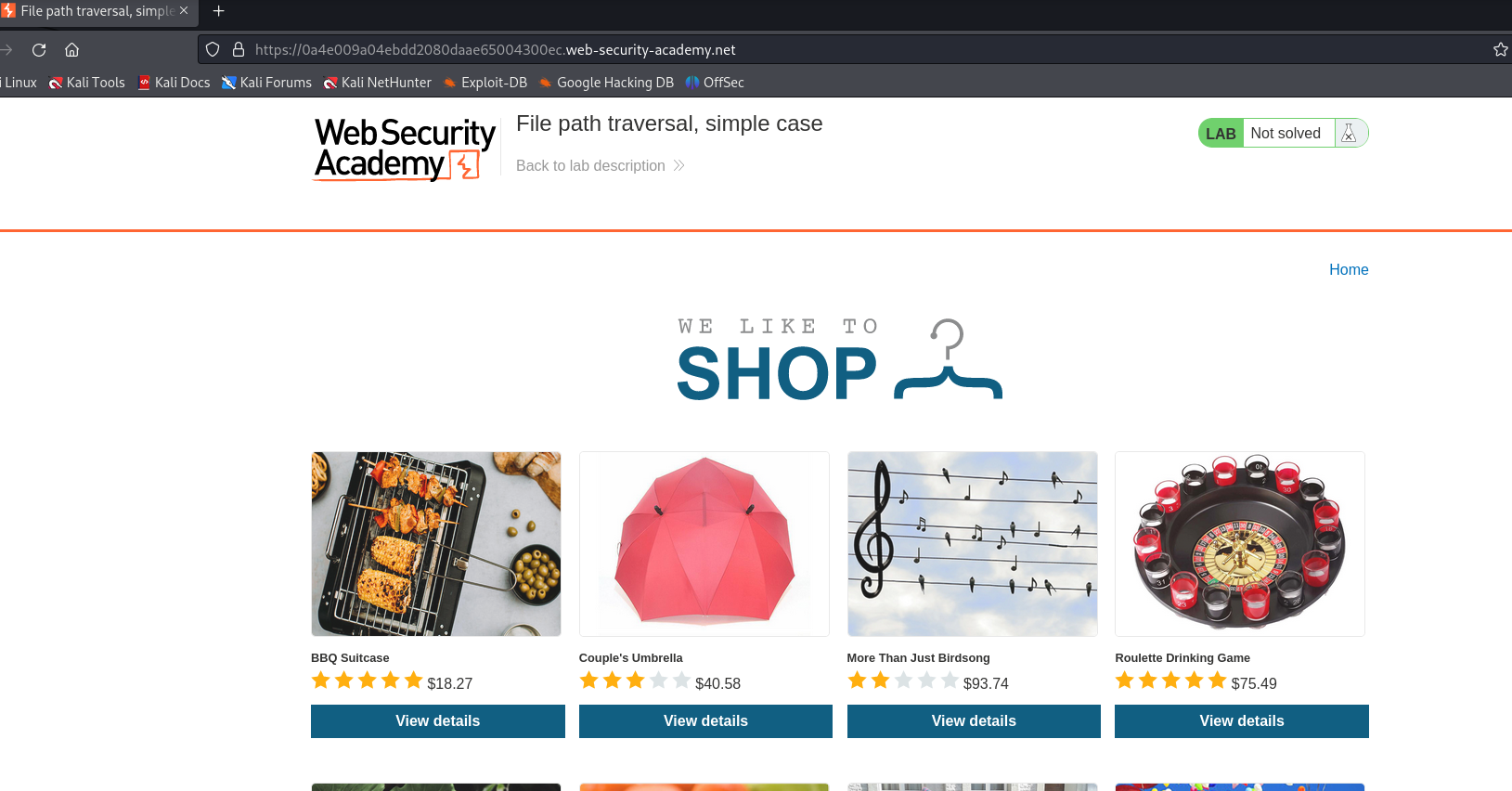
* **Application code and data**: Attackers can access the source code and data files of the application, potentially exposing sensitive business logic or user data.
* **Credentials for back-end systems**: Attackers might find configuration files containing usernames, passwords, or API keys used to access databases and other back-end systems.
* **Sensitive operating system files**: Critical system files such as /etc/passwd on Unix-based systems or C:\Windows\System32\config on Windows can be exposed, potentially revealing system configurations and user information.

**Walkthrough: File path traversal**

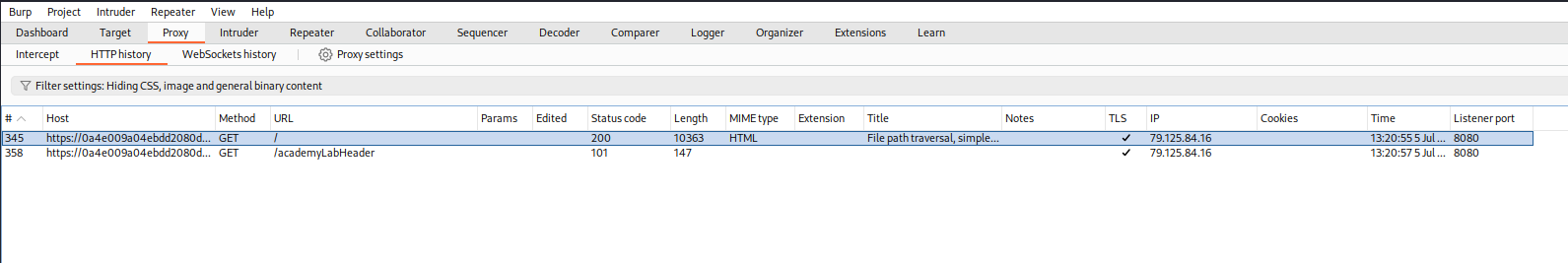
The application contains a path traversal vulnerability in the display of product images.

To solve the lab, retrieve the contents of the /etc/passwd file.

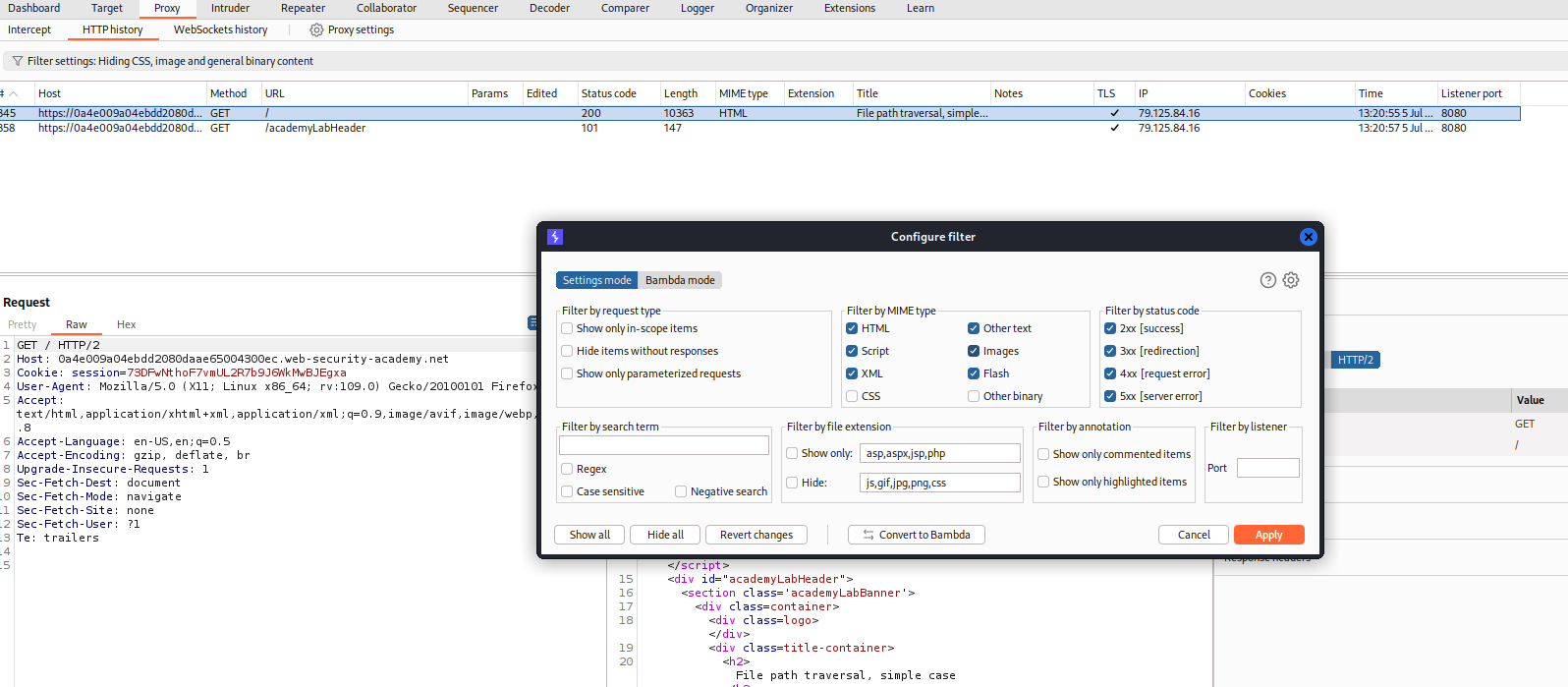
1. Send the **request** that fetches a product image to the **repeater**.
2. Open the homepage of the application.



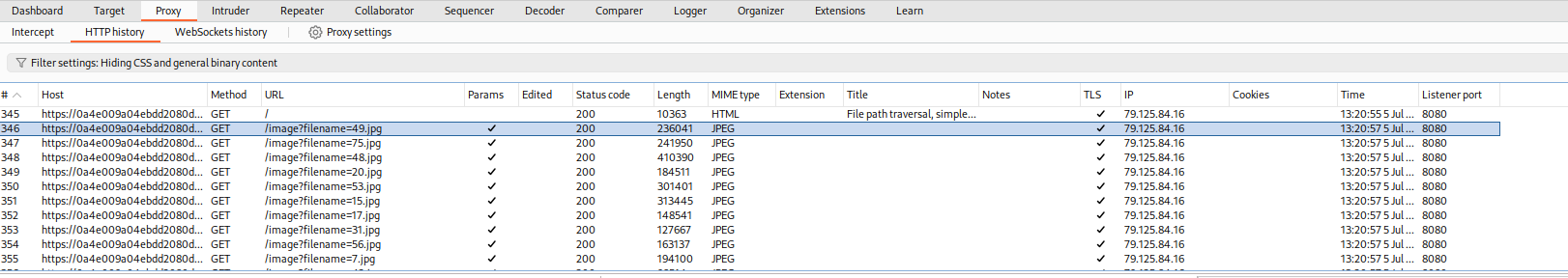
1. In burp suite open proxy tab, and go to HTTP history.



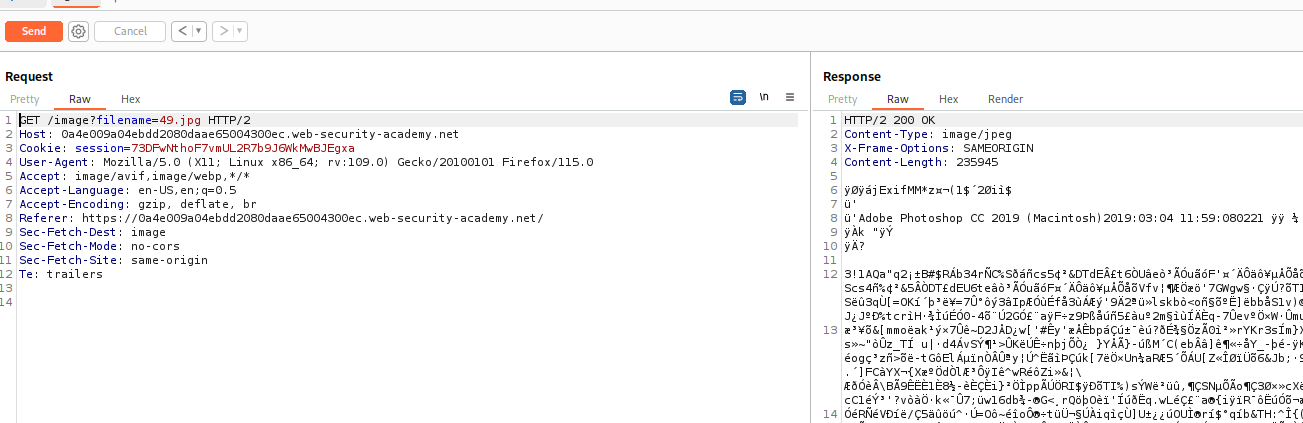
1. If you do not see any image requests, click on **Filter settings.** In **Filter by MIME Type** tick the image checkmark.



1. Now you will be able to see the images request.

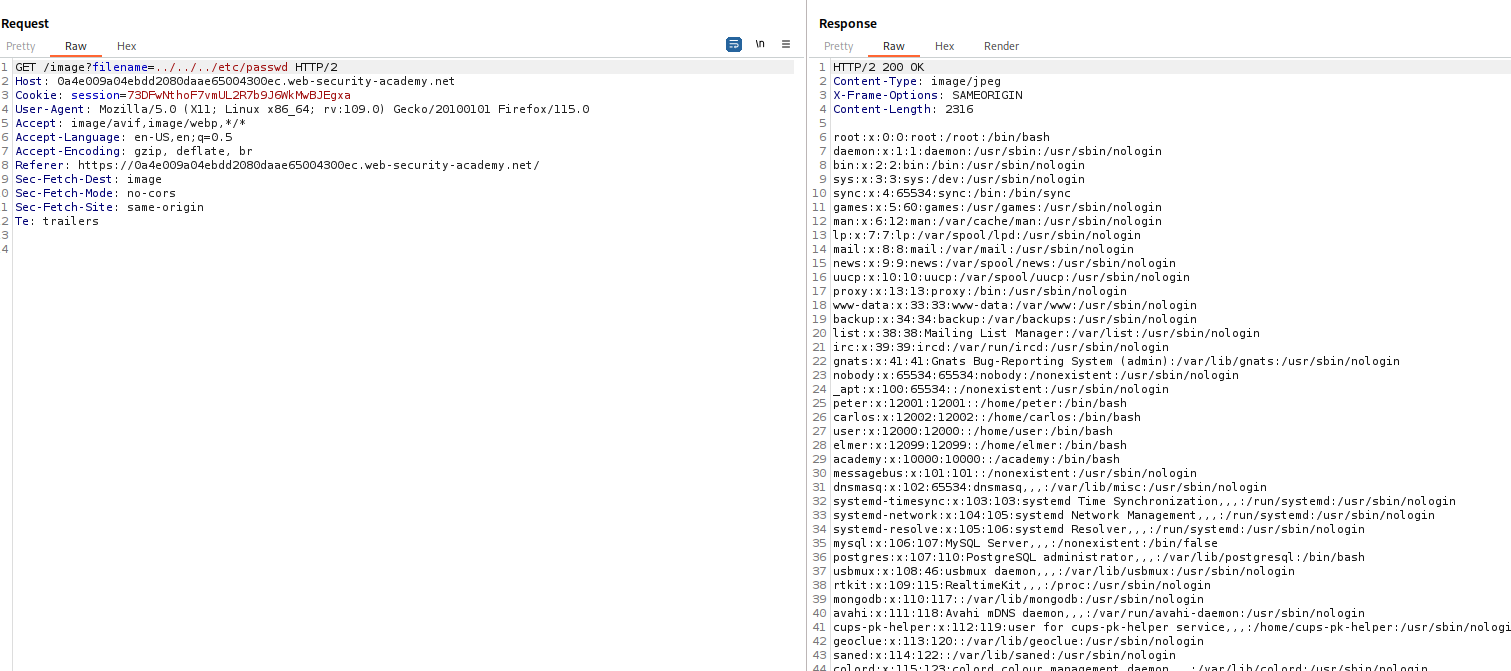


1. Send any image request to the repeater.



1. Modify the **filename** parameter, giving it the value:

**../../../etc/passwd**

****

Now we can observe in the response – all the contents of **/etc/passwd** file.

To prevent path traversal vulnerabilities, avoid passing user-supplied input to filesystem APIs. If necessary, validate input against a whitelist or permitted characters, canonicalize the path using filesystem APIs, and ensure it starts with the expected base directory. This layered defense minimizes the risk of unauthorized file access.