Lab 18	
Name	Dang Hoang Nguyen
Student ID	SE171946

What is a Path Traversal vulnerability, and how does it differ from other types of web application vulnerabilities? Explain the concept of path traversal, including how attackers can manipulate input to access or manipulate files outside of the intended directory.

Consider a website where users can submit photos for their profiles. It should ideally save these photographs in a specified directory, such as /images/profiles/. On the other hand, a malicious actor might create a path that goes outside the intended directory and access unauthorized files or even run system commands if the application fails to adequately evaluate user-supplied data. This is the fundamental vulnerability of a path traversal.

Attackers utilize certain characters and sequences to their advantage to change the file path that the application uses. Here's a closer look at typical methods:

1. **Dot-dot-slash (../):** This sequence represents moving up one directory level. For example, `../../../../etc/passwd` attempts to access the `/etc/passwd` file, which holds sensitive system information.
2. **Null Byte (\x00):** In some systems, a null byte terminates a string. So, an attacker might use `image.jpg\x00..\/etc\/passwd` to inject the path to `/etc/passwd` after the null byte, effectively bypassing validation checks.
3. **Directory Traversal Variations:** Depending on the operating system and web server configuration, other characters like `.` (current directory), `~` (home directory), or even spaces can be used for traversal.
4. **Double Encoding:** Attackers might encode special characters twice (e.g., `%2e%2e` for `../`) to bypass basic filters.
5. **Case Sensitivity:** Some systems are case-sensitive, so attackers might try variations like `PhOtO.jPg` to bypass case-insensitive filters.

Describe the process for exploiting a Path Traversal vulnerability in a web application. What are the common methods or techniques used by attackers to manipulate file paths, and how can these methods compromise the security of a server?

Process of Exploiting a Path Traversal Vulnerability:

- Identification: Attackers start by locating the locations where file paths are created using user input. This could include picture paths, file upload forms, or any other feature that requires accessing files in response to user input.
- Payload Creation: After that, they create a malicious payload that makes use of route traversal methods. This could be any input that modifies the file path that the program uses, such as a filename, URL parameter, or other input.
- Triggering the Vulnerability: In an attempt to gain unauthorized access, the attacker hopes to trigger the path traversal vulnerability by sending the forged payload to the application.

- In the event of success, the attacker can get access to files located outside of the designated directory, which could result in the exfiltration of data, the execution of code, or the total takeover of the system.

Common Methods and Consequences:

- Data Exfiltration: Via the theft of sensitive data, such as user information, configuration files, or source code, attackers might violate privacy and pilfer intellectual property.
- Malware Injection: By uploading malicious scripts onto the server, they may be able to compromise the whole system and spread the infection to more users.
- System Takeover: In severe circumstances, hackers may be able to access and alter vital system files to take total control of the server.
- Website Defacement: They might alter material on websites or send visitors to malicious websites, harming the credibility of the website and the trust of its users.

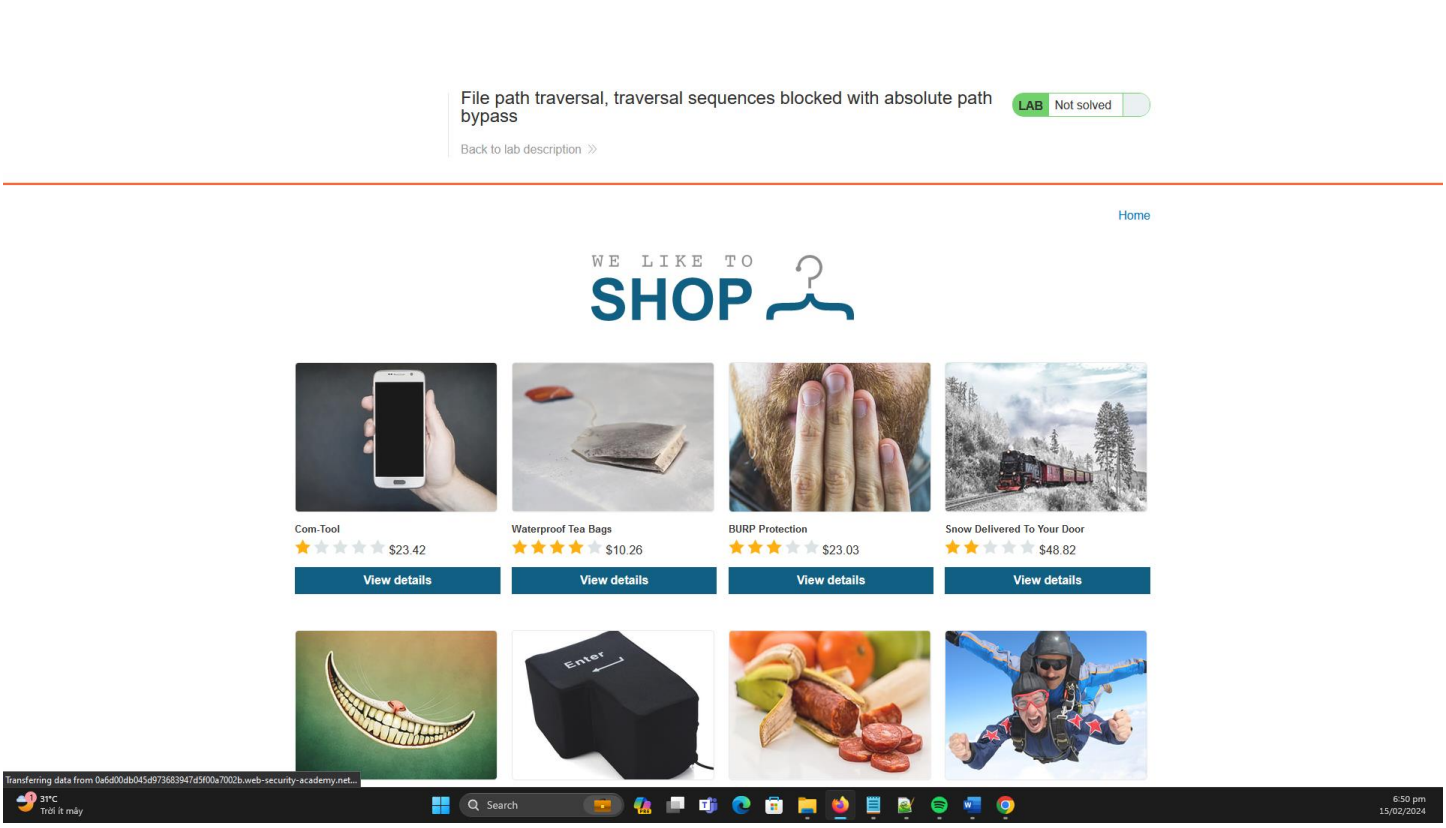
How Path Traversal Differs from Other Vulnerabilities:

While path traversal shares some similarities with other web application vulnerabilities, there are key distinctions:

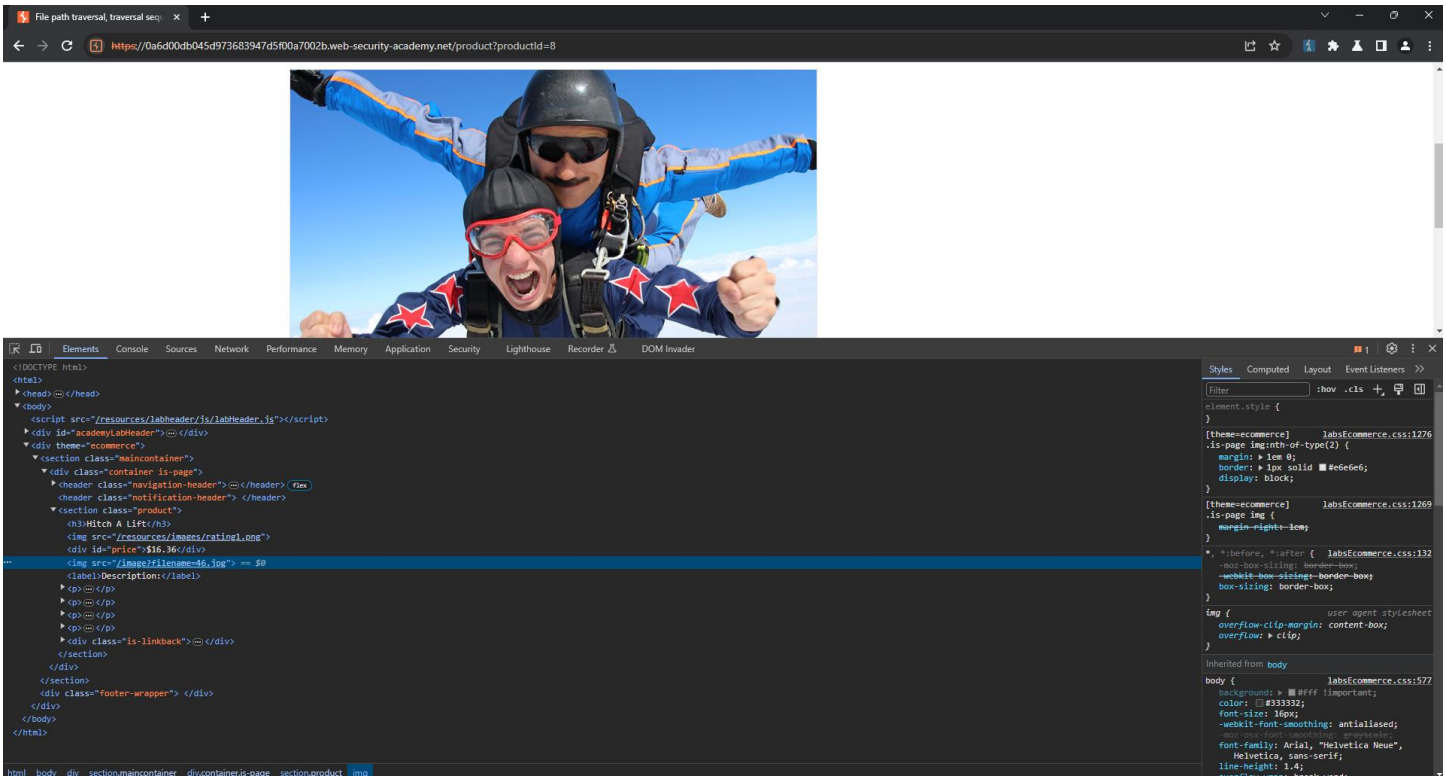
- Injection vs. Exploitation: Path traversal takes advantage of preexisting capabilities to gain unauthorized access to files, in contrast to SQL injection or XSS, which involve the insertion of malicious code.
- Impact Range: While certain vulnerabilities may only affect particular capabilities, path traversal may possibly affect the server filesystem as a whole.
- Stealthiness: Since path traversal attacks frequently don't require user input or produce error warnings, they might be challenging to identify.

Lab: File path traversal, traversal sequences blocked with absolute path bypass

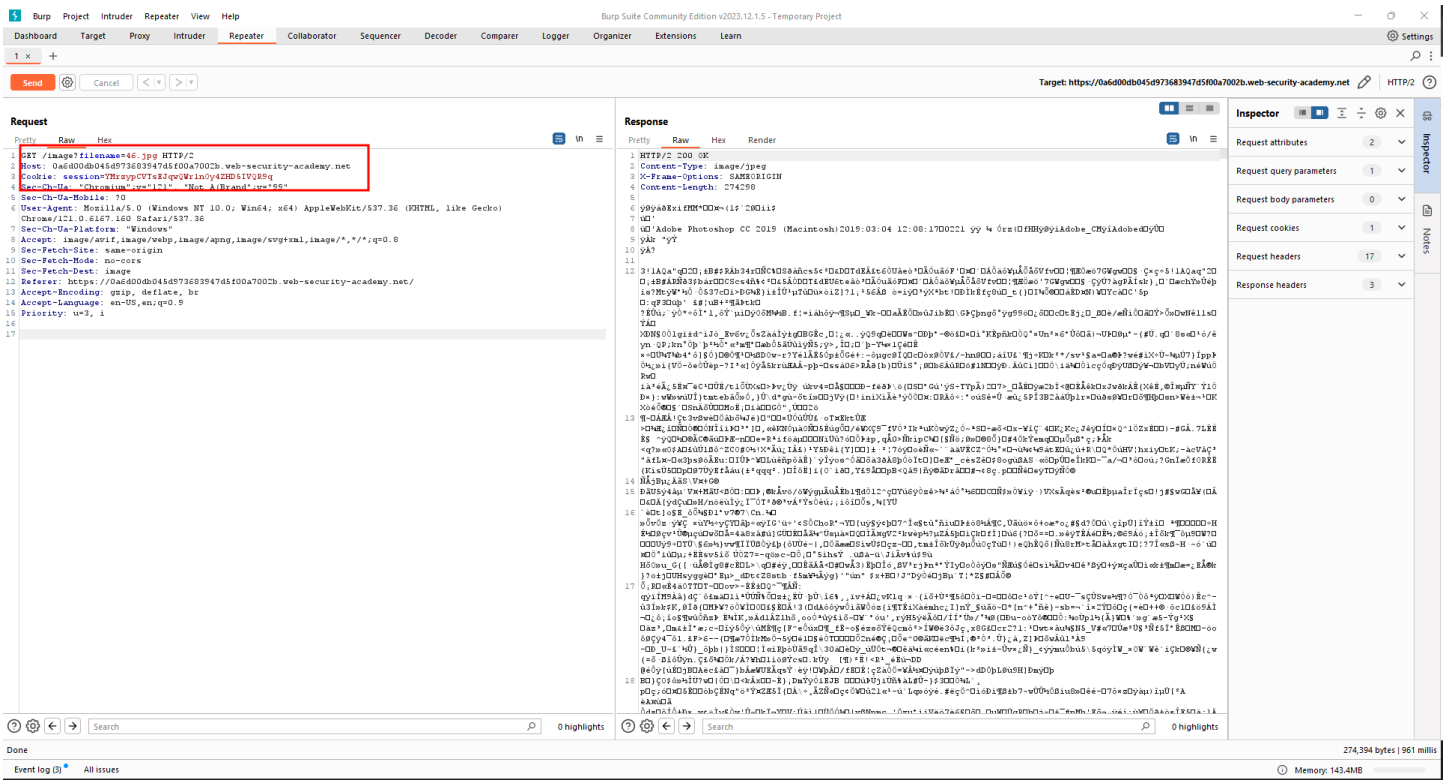
We come across a store website advertising several things as soon as we enter the lab. It's important to note that since PortSwigger Labs can change the material with every lab session, the products could not exactly match the screenshot. Thus, don't be alarmed if your things appear different.



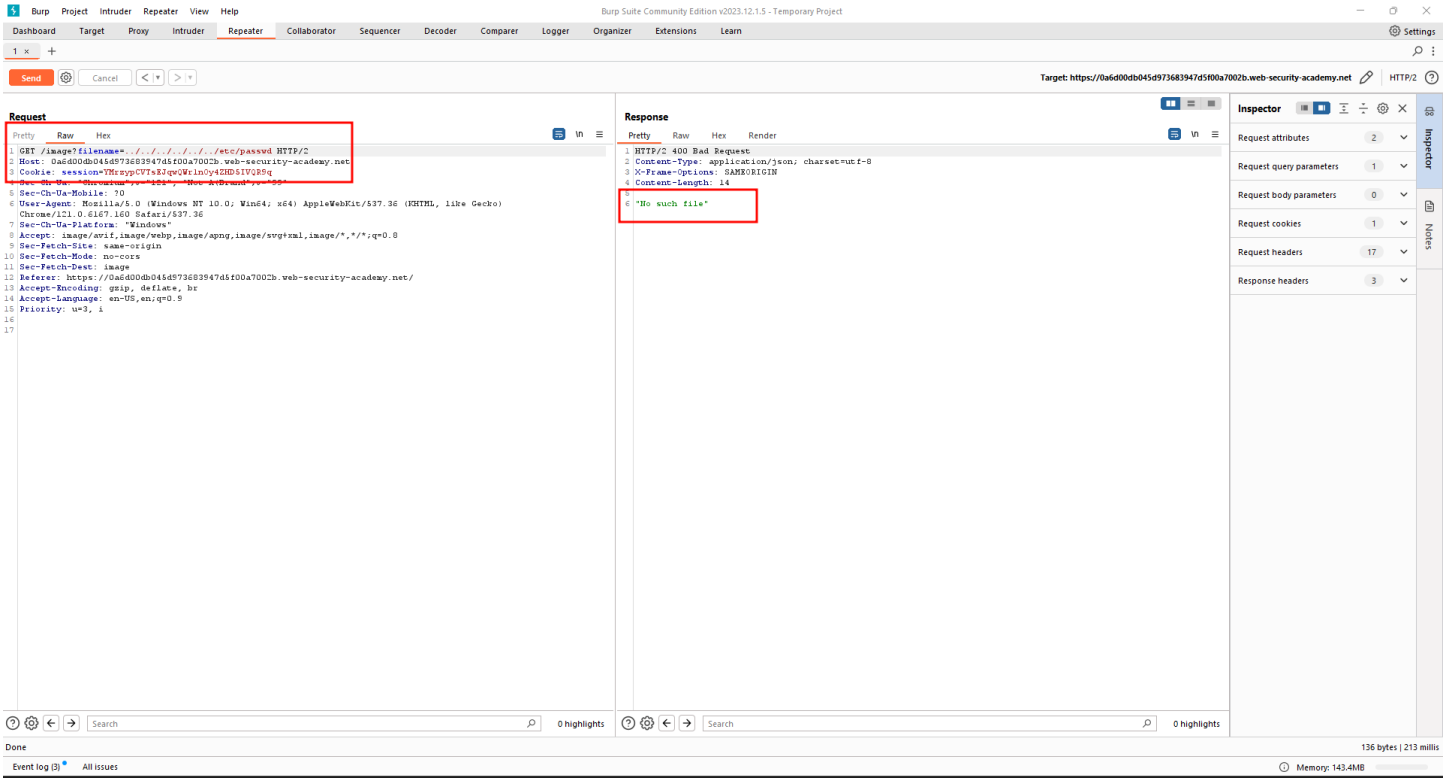
When selecting a product it opens a page containing comprehensive details about the product. Inspecting the source code we can observe an <img> tag referencing an image file named 46.jpg.



By using Burp Suite Repeater to process this request, we can manually try to attack the filename parameter.



In our pursuit of directory traversal our initial target is the /etc/passwd file, which is typically accessible to all users on a Linux system. To begin the attack it's important to know the behavior of the 'filename' parameter. Given that it references an image file on the system (in this case, a JPG), we can reasonably assume that this parameter reads files and displays their contents in the browser. We attempted to use the same payload as in our previous lab, 'File path traversal, simple case,' which was ../../../../../../etc/passwd. Unfortunately, we encountered an issue. It appears that the application is blocking the use of ../ in our payload.



We can make an attempt to access the file using an absolute path. An absolute path directly references a file without utilizing any traversal sequences. In this case, the payload would appear as /etc/passwd.

Our payload was successful and we have gained access to read the contents of the `/etc/passwd`.