

Root-Me Write-up: Directory Traversal

The screenshot shows a challenge card for 'Directory traversal'. It includes the following details:

- Points:** 25 Points
- Category:** Photo gallery v 0.01
- Author:** g0uZ, 31 July 2011
- Level:** Level ① (represented by a bar with 3 green segments)
- Validations:** 37440 Challengers, 10%
- Note:** 5 stars, 1301 Votes, with 'I like' and 'I don't like' buttons
- Statement:** Find the hidden section of the photo galery.
- Action:** A blue 'Start the challenge' button.

Initial Assessment

The challenge appeared to involve a **path traversal vulnerability**, which occurs when user-controlled input is not properly sanitized, allowing an attacker to access files or directories outside the intended scope by manipulating path parameters (e.g., `..`/).

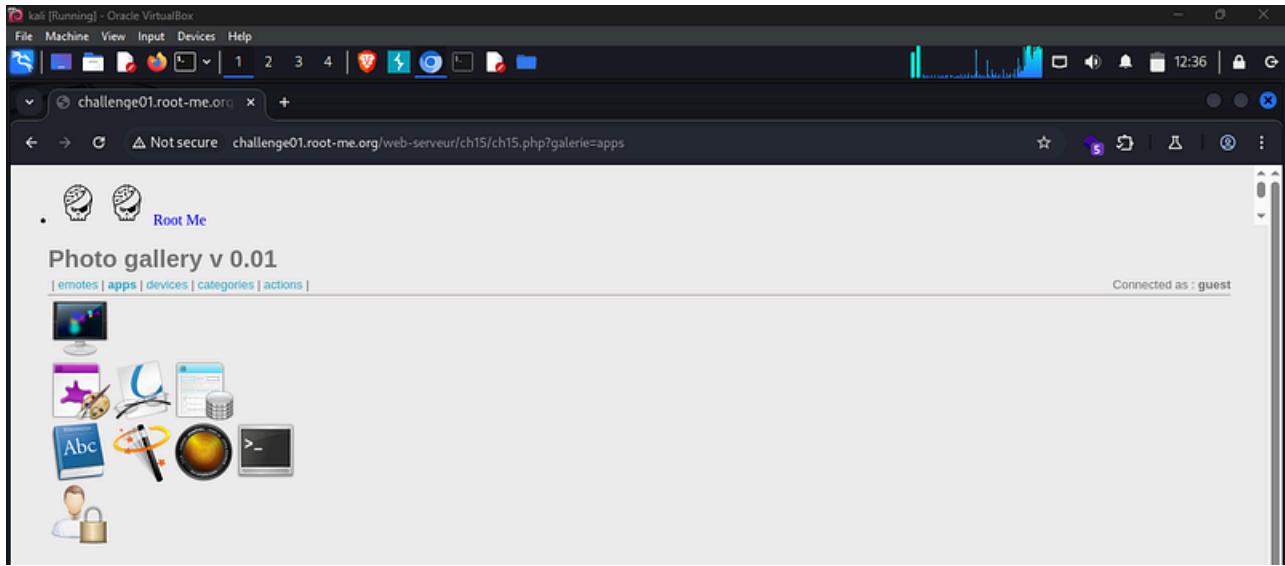
I began with basic test cases such as `..`/ and other typical traversal payloads to observe how the application handled directory navigation.

Enumeration

Next, I attempted directory enumeration using **Gobuster**, targeting commonly known directories. While this provided general context, the more useful insight came from manually inspecting the application.

Upon interacting with the gallery feature, I noticed that selecting different categories triggered requests of the following format:

```
/ch15.php?galerie=<category>
```



Based on the challenge hints, it was suggested that something important was hidden within the galerie parameter.

Exploitation

I tested the following payload:

```
/ch15.php?galerie=/
```

This effectively forced the application to reference another directory. As a result, a uniquely named image file was revealed:

86hwnX2r

The HTML source showed the following element:

```

```

Using the browser's inspector, I examined the full path and then attempted to access it directly:

The screenshot shows a browser window with the URL `challenge01.root-me.org/web-serveur/ch15/ch15.php?galerie=/.` The page content includes a table with two rows. The first row has one column with an image of a folder named "86hwnX2r". The second row has two columns, each with an image of a folder named "emotes" and "apps". The developer tools are open, showing the DOM structure and the CSS styles applied to the images.

/ch15.php?galerie=86hwnX2r

This led to another discovery. The page referenced a text file:

```

```

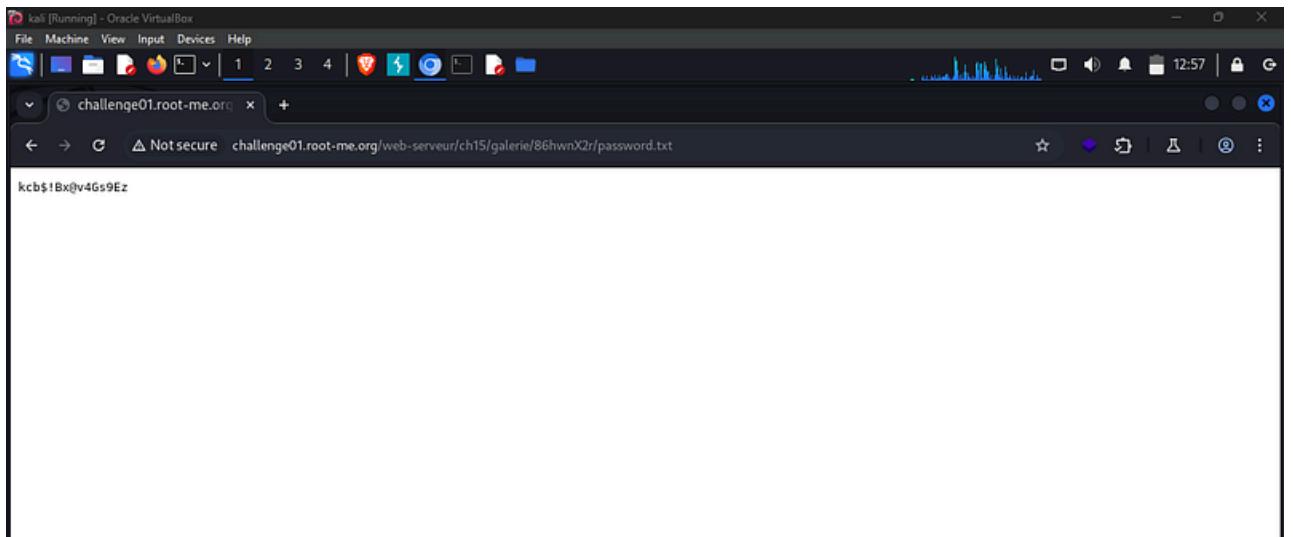
The screenshot shows a browser window with the URL `challenge01.root-me.org/web-serveur/ch15/ch15.php?galerie=86hwnX2r|`. The page title is "Photo gallery v 0.01". It contains a table with three rows. The first row has one column with an image of a folder named "password.txt". The second row has two columns, both empty. The third row has two columns, both empty. The developer tools show the HTML structure and the CSS styles applied to the images.

Flag Retrieval

From the src attribute, the full file path could be inferred. I accessed it directly via:

/ch15/galerie/86hwnX2r/password.txt

This successfully revealed the flag.



Flag

kcb\$!Bx@v4Gs9Ez

By [Alexander Sapo](#) on [December 30, 2025](#).

[Canonical link](#)

Exported from [Medium](#) on February 7, 2026.