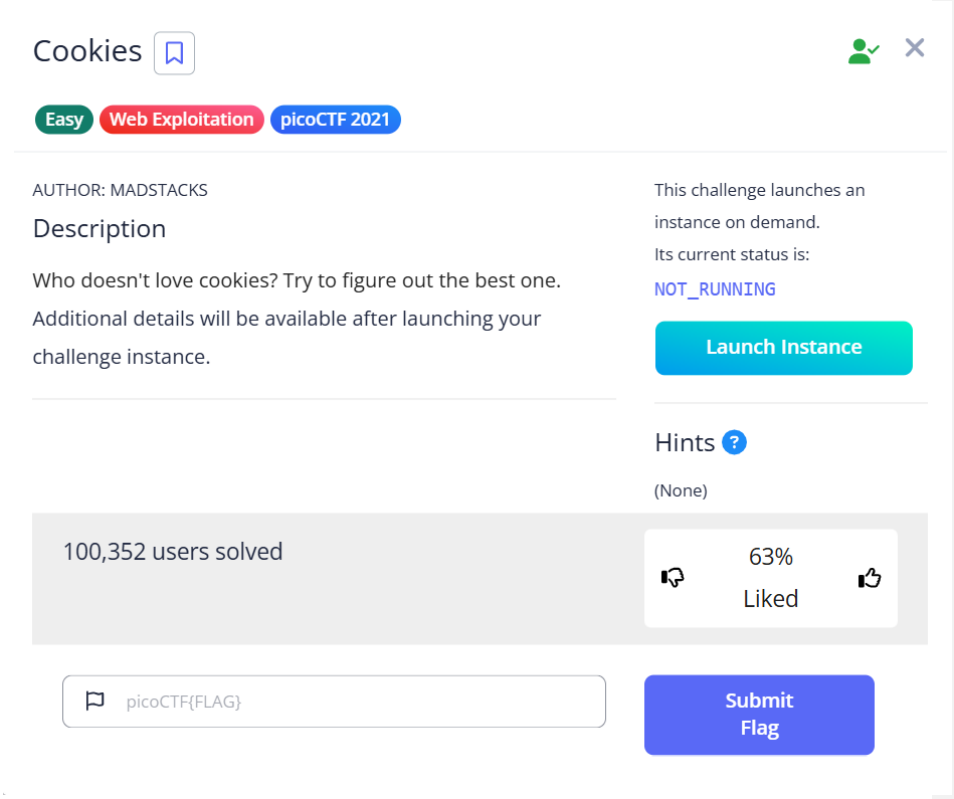
Saint Columban College
College of Computing Studies
Pagadian City

Cookies



Author: The Analyst: Hyposelenia

Challenge: Cookies

Category: Web Exploitation

Date: 02/15/26

I. Objective

The objective of this challenge is to understand how browser cookies work, how they can be manipulated, and how improper handling of cookies in web applications can lead to security vulnerabilities that expose sensitive information such as hidden flags.

II. Background

Cookies are small pieces of data stored by websites in a user's browser. They are commonly used to remember user preferences, session information, or progress within a website. While cookies are helpful, they can become a security risk if a website **trusts user-controlled cookies without proper server-side validation**.

In web exploitation challenges, cookies are often used to demonstrate how attackers can modify client-side data to trigger unintended behavior in a web application. This challenge focuses on recognizing patterns in cookie values and exploiting weak logic tied to those values.

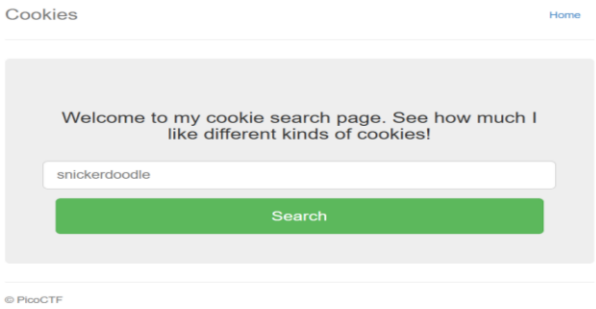
III. Tool Used

- **Web Browser (Google Chrome)** – Used to access the challenge website
- **Browser Developer Tools (Inspect)** – Used to view and modify cookies

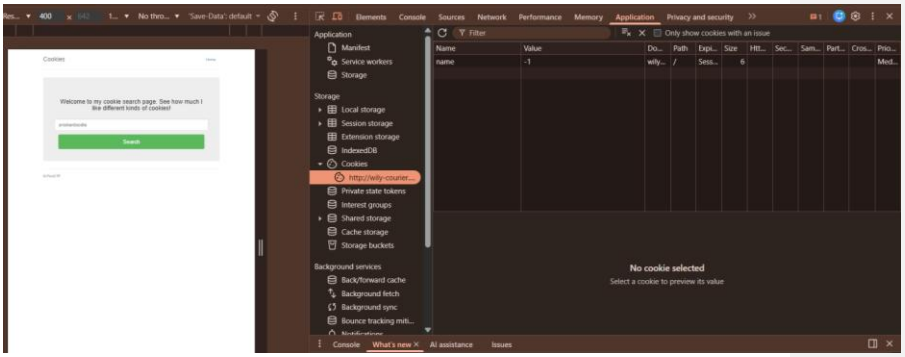
IV. Methodology

1. Launched the challenge instance.

2. Copied the provided URL and accessed it using a web browser.
3. Explored the website and observed the search input field.

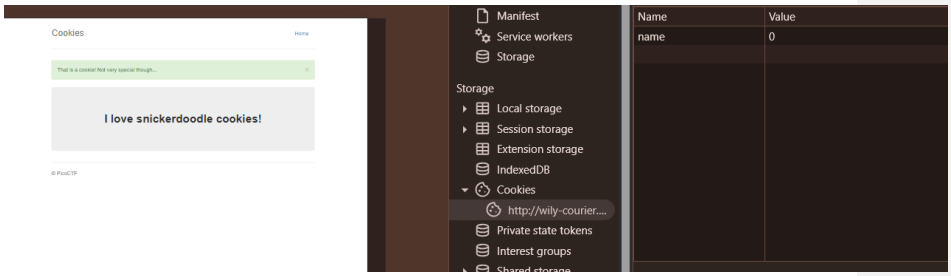


4. Noticed that the challenge title referenced “Cookies,” suggesting the use of browser cookies.
5. Right-clicked the page and opened **Inspect (Developer Tools)**.
6. Navigated to the **Application** tab and selected **Cookies** for the website.



7. Observed a cookie where the name value was initially set to -1, even before any input was provided.

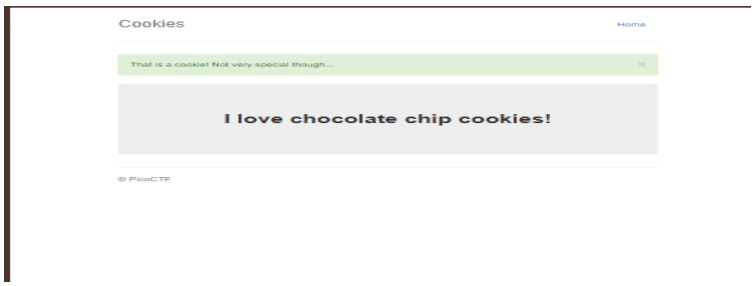
8. Entered the word **"snickerdoodle"** into the search box and clicked the search button.
9. Noticed that the cookie value changed from -1 to 0.



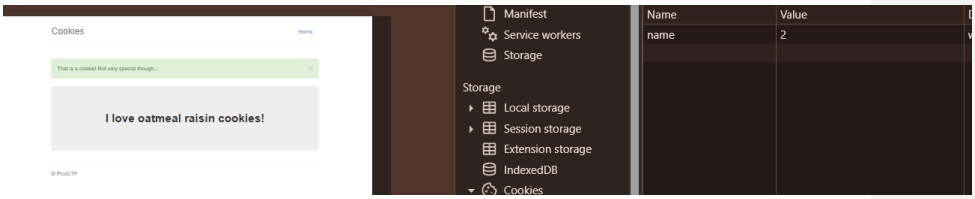
10. Manually modified the cookie value from 0 to 1 and refreshed the page.

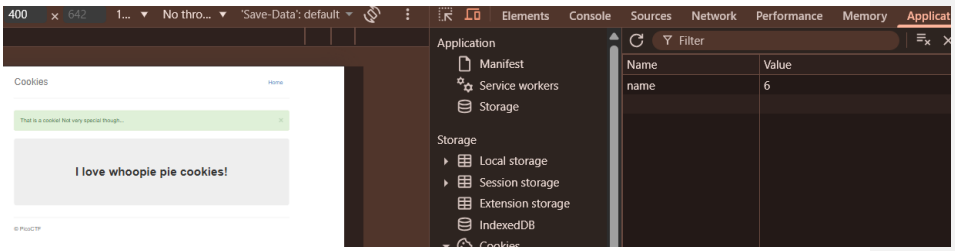
11. Observed that the displayed cookie name on the site changed.

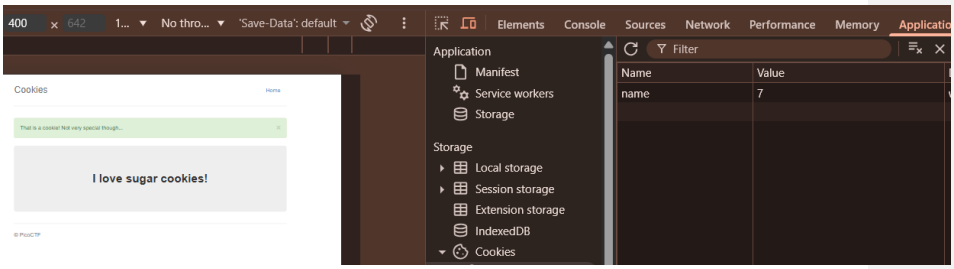
Commented [Z.1]:



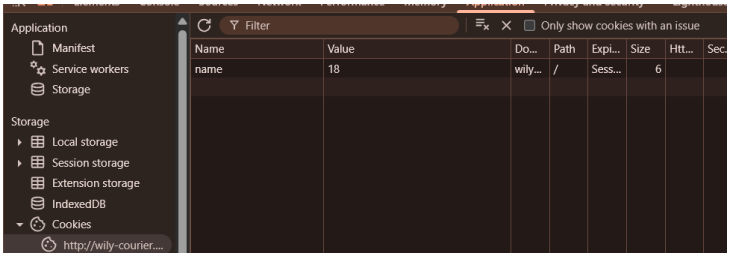
12. Continued incrementing the cookie value (2, 6, 7, and onward), refreshing the page each time.
13.



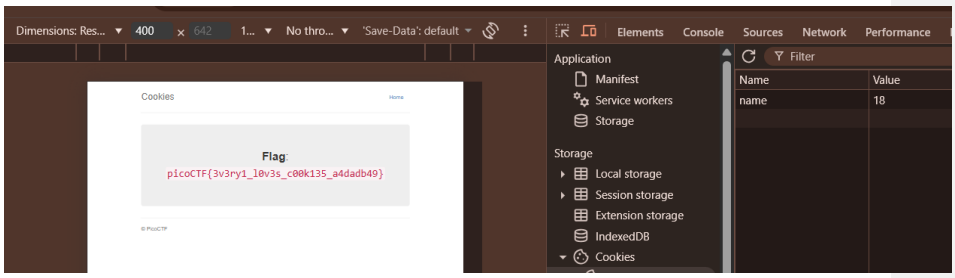




14. Reached a cookie value of **18** and refreshed the page.



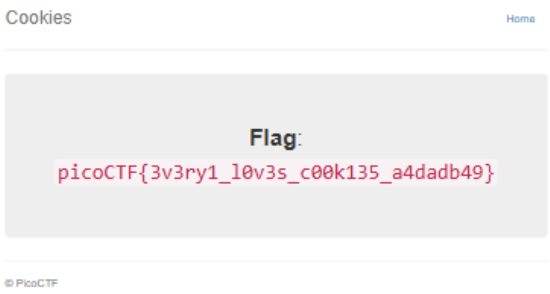
15. Observed that instead of a cookie name, the **flag was displayed directly on the webpage**.



16. Retrieved and submitted the flag.

V. Result

picoCTF{3v3ry1_l0v3s_c00k135_a4dadb49}



VI. Explanation

Think of a cookie like a **sticky note** a website puts in your browser. The sticky note might say:

- *"You are on cookie number 0"*
- *"You are on cookie number 1"*
- *"You are on cookie number 2"*

The website reads this sticky note to decide what to show you.

In this challenge, the website **trusted the sticky note completely**. By changing the number on the note, we could tell the website to jump ahead—until it accidentally showed the secret flag.

Now you wondered, what did the cookie did in this challenge?

- *The cookie value acted like a **counter or index***
- *Each number corresponded to a different message or cookie name*
- *When the value reached **18**, the application revealed the flag*

This means the website relied entirely on **client-side cookie values** to control behavior.

Now, Cookies are stored in the browser, which means:

- Users can view them
- Users can modify them
- Attackers can exploit them

If a website uses cookies to control sensitive logic (such as access levels, secrets, or progression), attackers can manipulate those cookies to bypass restrictions.

I'll give you a real-life scenario:

Imagine logging into Facebook on a **public computer**:

- Facebook stores your session in a cookie
- If you forget to log out, the cookie remains
- Another person opens Inspect, reuses your session cookie
- They access your account **without knowing your email or password**

This is why cookies must be:

- Properly secured
- Expired correctly

- Validated on the server side

So, the purpose of the challenge?

This challenge teaches:

1. How cookies work
2. How cookies can be inspected and modified
3. Why trusting client-side data is dangerous
4. How attackers exploit weak cookie logic

This is a classic example of:

Insecure Client-Side State Management

VII. Conclusion

The Cookies challenge demonstrates how improper use of browser cookies can expose sensitive information. By manipulating a cookie value that controlled application behavior, the flag was revealed without exploiting any complex vulnerabilities. This highlights the importance of server-side validation, secure session handling, and cautious use of cookies in web applications. Understanding cookie-based vulnerabilities is essential for both secure web development and effective web exploitation.

— **The Analyst: Hyposelenia**