CTF Report: Finding a Hidden Flag with the Web Inspector

Ayoub Goubraim

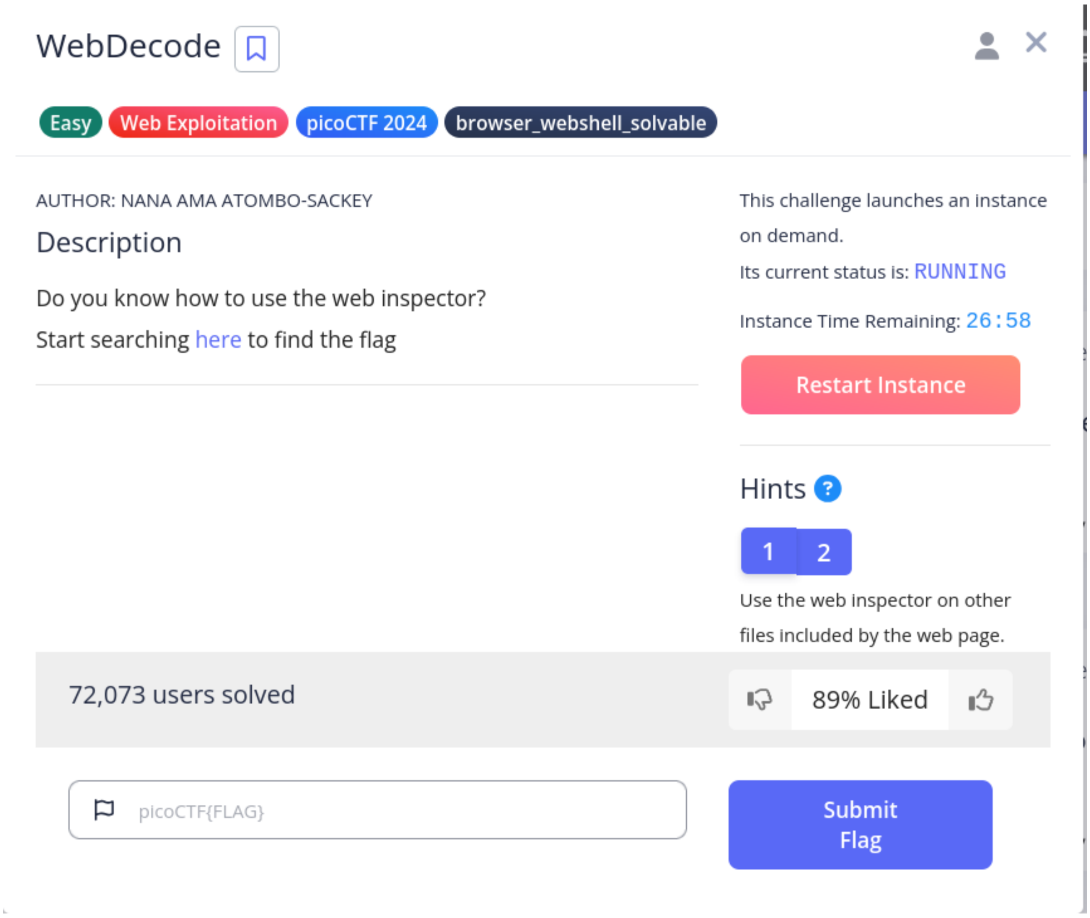
September 9, 2025

Abstract

This report documents the solution to the picoCTF challenge **WebDecode**. Using only a browser's Developer Tools (Web Inspector), we enumerate static pages, examine embedded attributes, discover a Base64-encoded token in the DOM, and decode it to obtain the flag.

1 Challenge Overview

The challenge statement hints at using the web inspector and exploring "other files included by the page." An on-demand instance hosts a simple static website.



WebDecode

AUTHOR: NANA AMA ATOMBO-SACKEY

Description

Do you know how to use the web inspector?

Start searching here to find the flag

This challenge launches an instance on demand.

Its current status is: RUNNING

Instance Time Remaining: 26:58

Hints

Use the web inspector on other files included by the web page.

72,073 users solved

89% Liked

Figure 1: Challenge page: *WebDecode*.

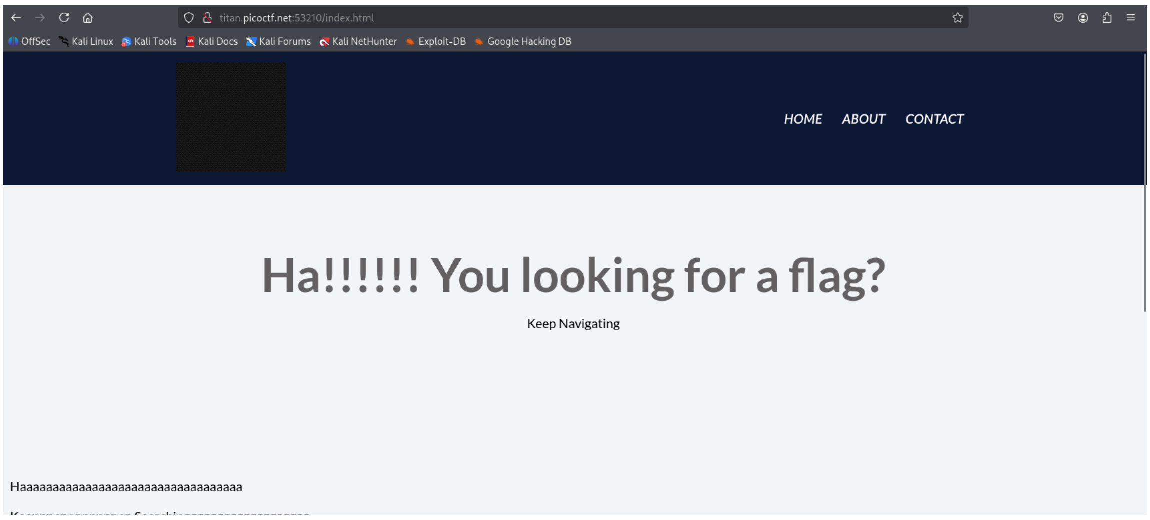
2 Environment & Tools

- **OS:** Kali Linux.

- **Browser:** Firefox.
- **Tools:** Built-in Web Inspector (`F12`), and CyberChef (for Base64 decoding).

3 Reconnaissance

Navigating to the instance reveals a small multi-page site with a navbar (`HOME`, `ABOUT`, `CONTACT`). The landing page encourages further navigation.



titan.picoctf.net:53210/index.html

Ha!!!!!! You looking for a flag?

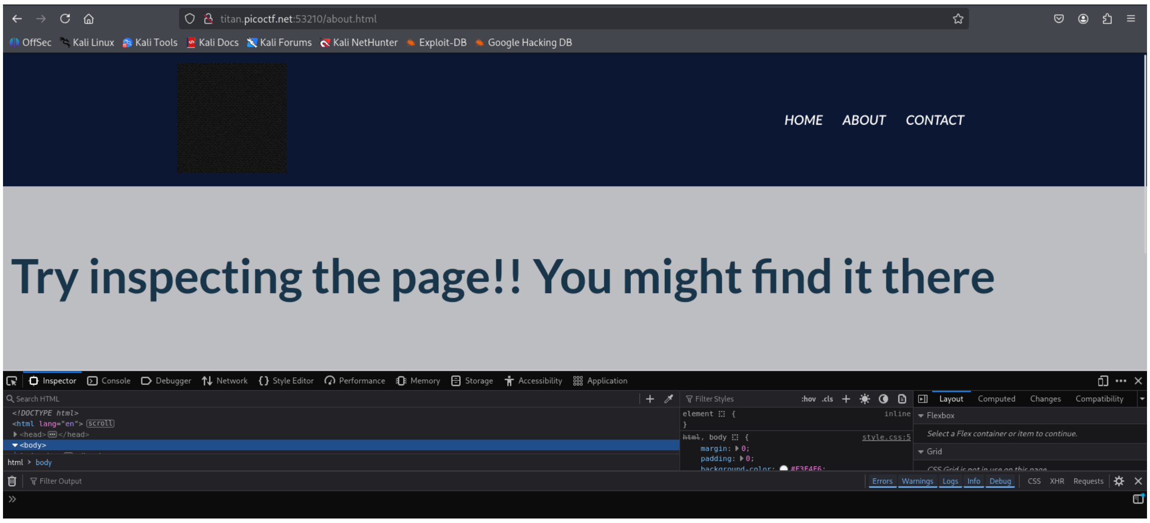
Keep Navigating

Haaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Figure 2: Homepage suggests continuing to navigate.

4 Inspecting the DOM

On the `about.html` page, a large banner explicitly suggests inspecting the page. Opening Developer Tools, the DOM shows a suspicious attribute embedded in a section element: `notify_true="<base64_stri`



titan.picoctf.net:53210/about.html

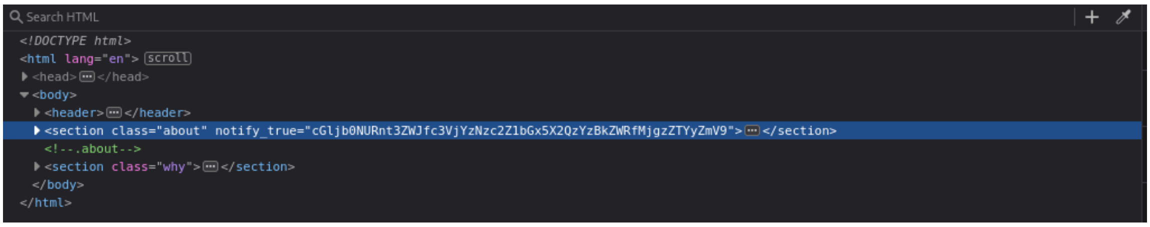
Try inspecting the page!! You might find it there

```
<!DOCTYPE html>
<html lang="en"> scroll
<head>…</head>
<body>
```

```
element {                inline
}
html, body {        style.css:5
    margin: 0;
    padding: 0;
    background-color: #F3F4F6;
```

Select a Flex container or item to continue.

Figure 3: About page with the Web Inspector open. The hint says to inspect the page.

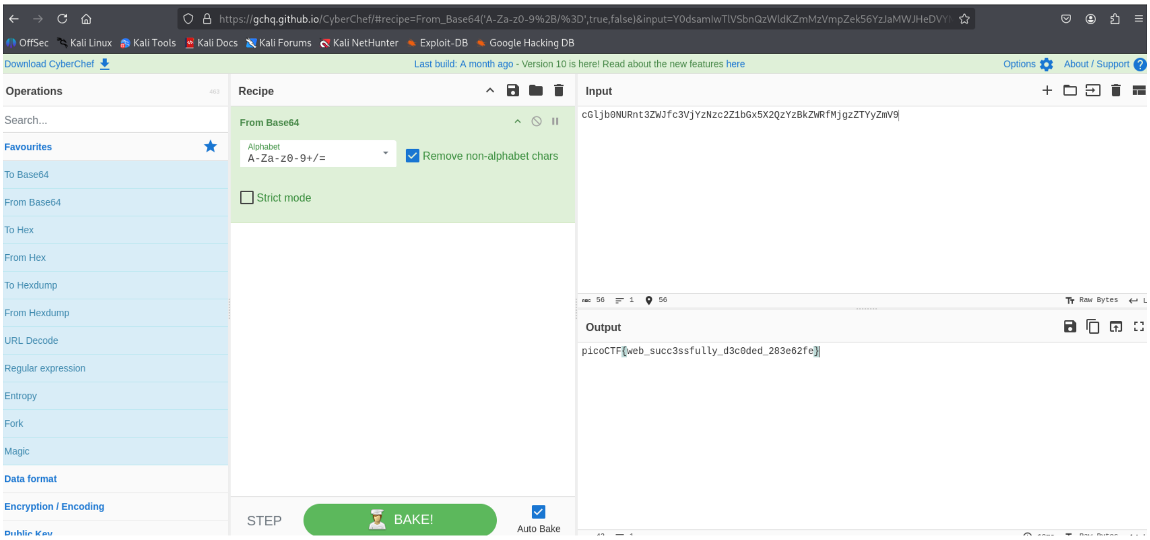


```
<!DOCTYPE html>
<html lang="en"> scroll
  <head>…</head>
  <body>
    <header>…</header>
    <section class="about" notify_true="cGljb0NURnt3ZWJfc3VjYzNzc2Z1bGx5X2QzYzBkZWRfMjgzZTYyZmV9">…</section>
    <!--.about-->
    <section class="why">…</section>
  </body>
</html>
```

Figure 4: The Base64-encoded value found in the DOM (`notify_true` attribute).

5 Decoding and Extracting the Flag

The attribute value is Base64. Decoding it (e.g., in CyberChef with the *From Base64* operation) yields the flag in the standard picoCTF format.



https://gchq.github.io/CyberChef/#recipe=From_Base64('A-Za-z0-9%2B/%3D',true,false)&input=Y0dsamIwTlVSbnQzWldKZmMzVmpZek56YzJaMWJHeDVY

From Base64
Alphabet
A-Za-z0-9+/=
☑ Remove non-alphabet chars
□ Strict mode

Input

```
cGljb0NURnt3ZWJfc3VjYzNzc2Z1bGx5X2QzYzBkZWRfMjgzZTYyZmV9
```

Output

```
picoCTF{web_succ3ssfully_d3c0ded_283e62fe}
```

Figure 5: Decoding the Base64 token reveals the flag.

6 Result

The decoded string is the challenge flag:

`picoCTF{web_succ3ssfully_d3c0ded_283e62fe}`

7 Discussion & Takeaways

This challenge reinforces common web-inspection techniques:

- Always check the DOM for hidden attributes, comments, or embedded data.
- Explore all linked pages (`about`, `contact`, assets).
- Recognize common encodings like Base64 and have a quick way to decode them (CyberChef or CLI tools).

8 Conclusion

By systematically navigating the site and inspecting the DOM, we identified an embedded Base64 string and decoded it to retrieve the flag, fulfilling the *WebDecode* challenge requirements.