



TARUC CTF – Season 1 (HIDE AND SEEK)

Write-ups

Team Name: xOrry

University Represented : APU

Team Leader: Wesley Wong Kee Han

Team Member: Lim Wei Xun, Ong Fo Seng

Flags Found: 8

Flag #01 (200)

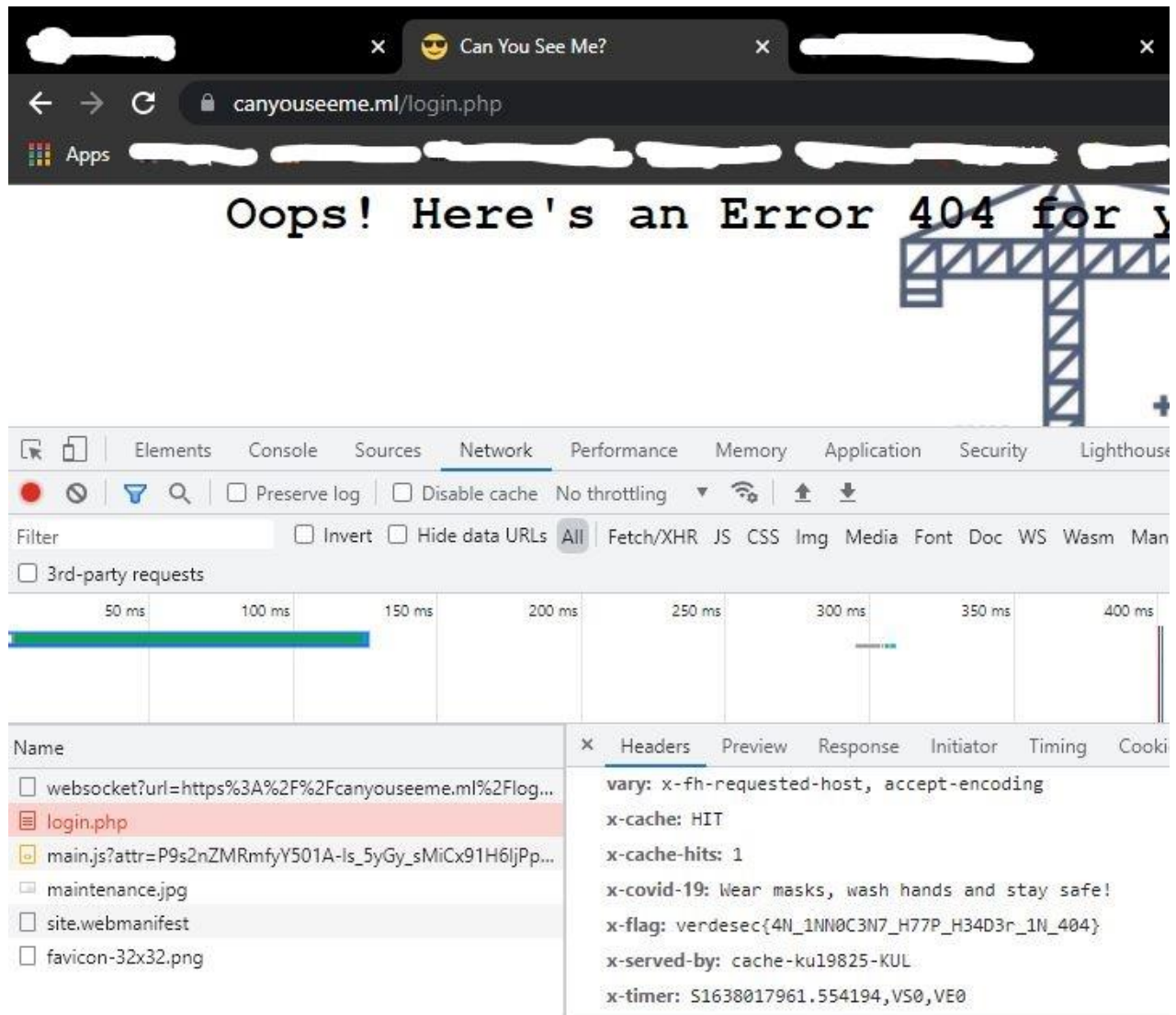
The login credentials sent to leader's email is a sussy baka. It looks like **base encryption** without the **==** padding at the end. After verifying, we managed to find our first flag.

The screenshot shows a web-based Base64 decoding tool. On the left, under the 'Recipe' tab, the 'From Base64' section is active. It shows a dropdown menu set to 'Alphabet' with the character set 'A-Za-z0-9+/' and a checked option for 'Remove non-alphabet chars'. The main area is split into 'Input' and 'Output' sections. The 'Input' section contains the string 'cGxheWVvMTQ3LChpbkZjTkY6bk4sdmVvZGVzZWw7N0gzX0YxcjU3X0ZMNDZ9'. The 'Output' section shows the decoded result: 'player147, (inFcNF:nN,verdesec{7H3_F1r57_FL46}'. Metadata for both input and output is displayed at the top of their respective sections.

Section	start	end	length	time	length	lines
Input	60	60	60			
Output	45	45	45	2ms	45	1

Flag #04 (712)

Since mass scanning tools are prohibited, we tried to search for hidden directories manually, such as `login.php`, `admin.php` and etc. Under the network tab, the server responded with code 404, which indicate unavailable resource. However, there is an attribute of `x-flag` in the header section that holds the flag.



Flag: `verdesec{4N_1NN0C3N7_H77P_H34D3r_1N_404}`

Flag #05 (986)

By inspecting the HTML source, we can see a suspicious string that exists as a developer's comment after expanding the head section.

```
<link rel="manifest" href="site.webmanifest">
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/mvp.css/1.7.4/mvp.min.css">
<link rel="stylesheet" href="style.css">
<!--T6P56Q64Yj_(0pt_&s0q_|o0P_sps`_)q0j&jA-->
```

By using a cipher analyzer, none of the results has a distinct and outstanding encryption method. Therefore, it is likely to be a combination of encryption methods.

The screenshot shows the dCode Cipher Identifier website. On the left, there's a search bar with the text "e.g. type 'caesar'" and a list of suggested ciphers: ASCII Shift Cipher, Keyboard Shift Cipher, Leet Speak 1337, and ROT-47 Cipher. On the right, the "ENCRYPTED MESSAGE IDENTIFIER" section shows the input "T6P56Q64Yj_(0pt_&s0q_|o0P_sps`_)q0j&jA" and an "ANALYZE" button. Below the input, there's a section for "CLUES/KEYWORDS (IF ANY)" and a link to "Symbols Identifier".

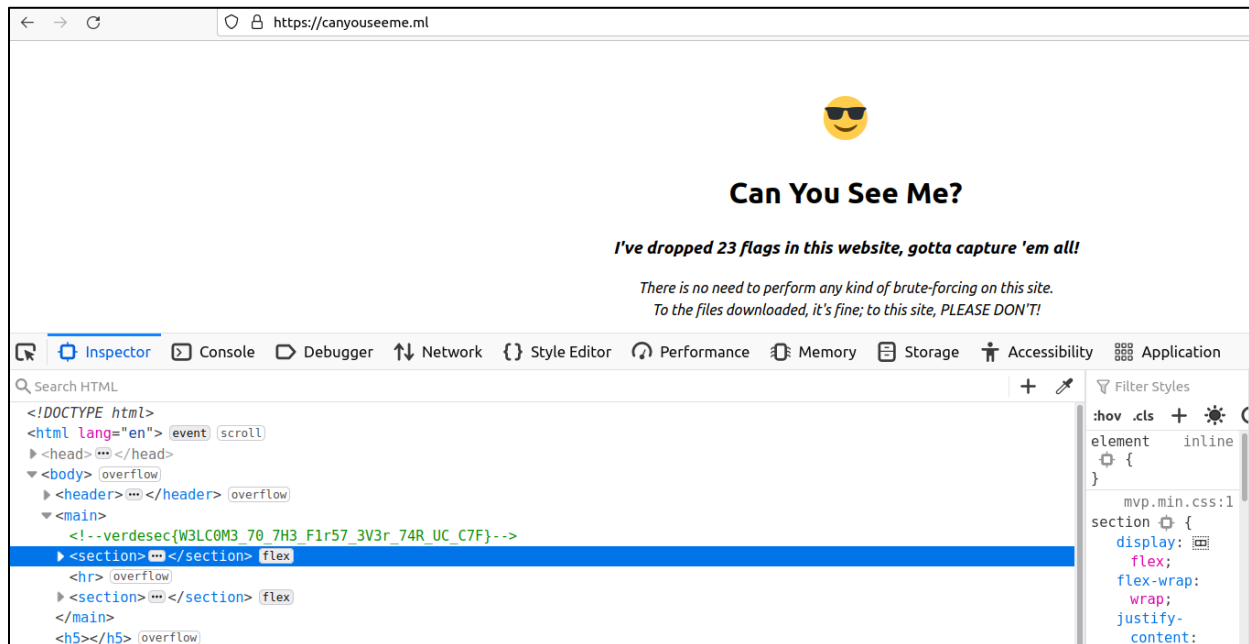
The flag can be found by using ROT 13 followed by ROT 47.

The screenshot shows the ROT13 and ROT47 encryption tool interface. The ROT13 section has checkboxes for "Rotate lower case chars" and "Rotate upper case chars", both checked, and a dropdown for "Amount" set to 13. The ROT47 section has a dropdown for "Amount" set to 47. The "Input" field contains the text "T6P56Q64Yj_(0pt_&s0q_|o0P_sps`_)q0j&jA" and the "Output" field shows the result "verdesec{H0W_480U7_50M3_r074710N5_HUH}".

Flag: verdesec{H0W_480U7_50M3_r074710N5_HUH}

Flag #06 (200)

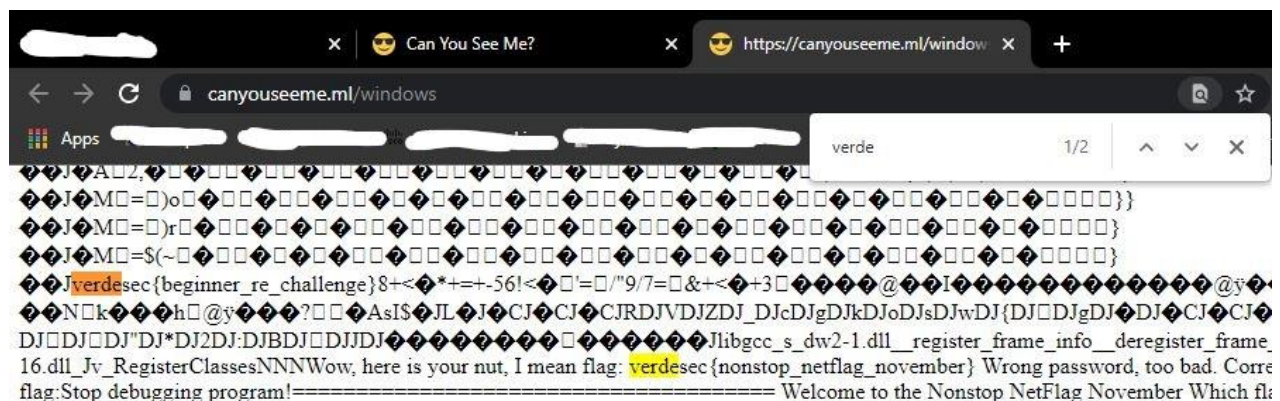
No explanation is needed. Flag in plaintext as source comment. EZPZ.



Flag: **verdesec{W3LC0M3_70_7H3_F1r57_3V3r_74R_UC_C7F}**

Flag #11 (200) and Flag #17 (200)

Simple query for flag format using chrome's search function CTRL+F.

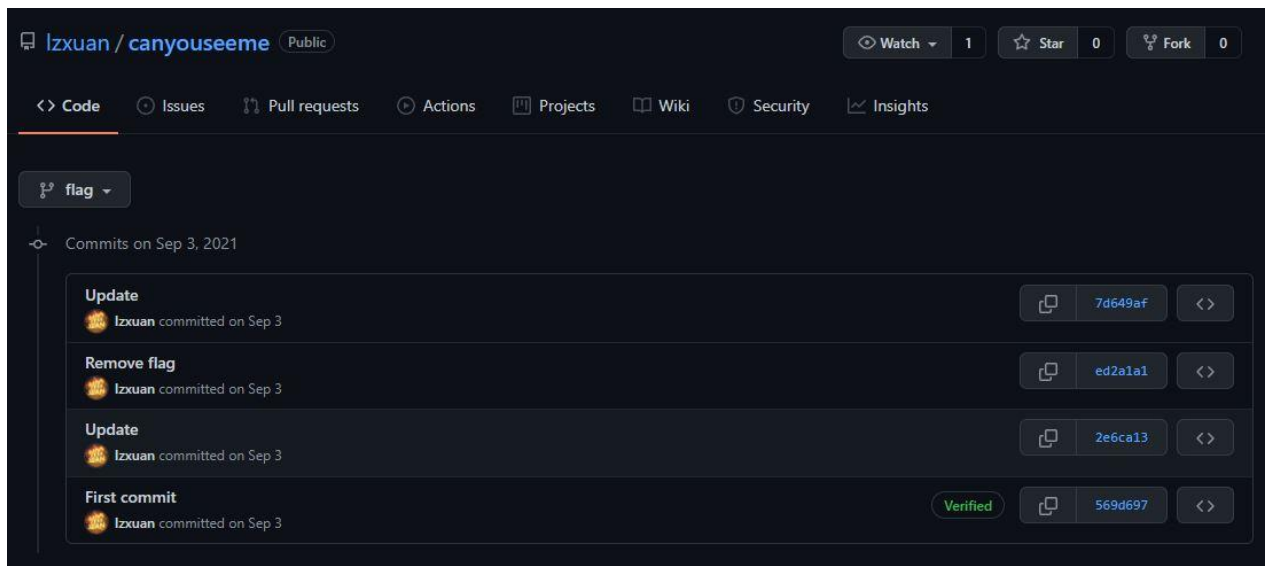
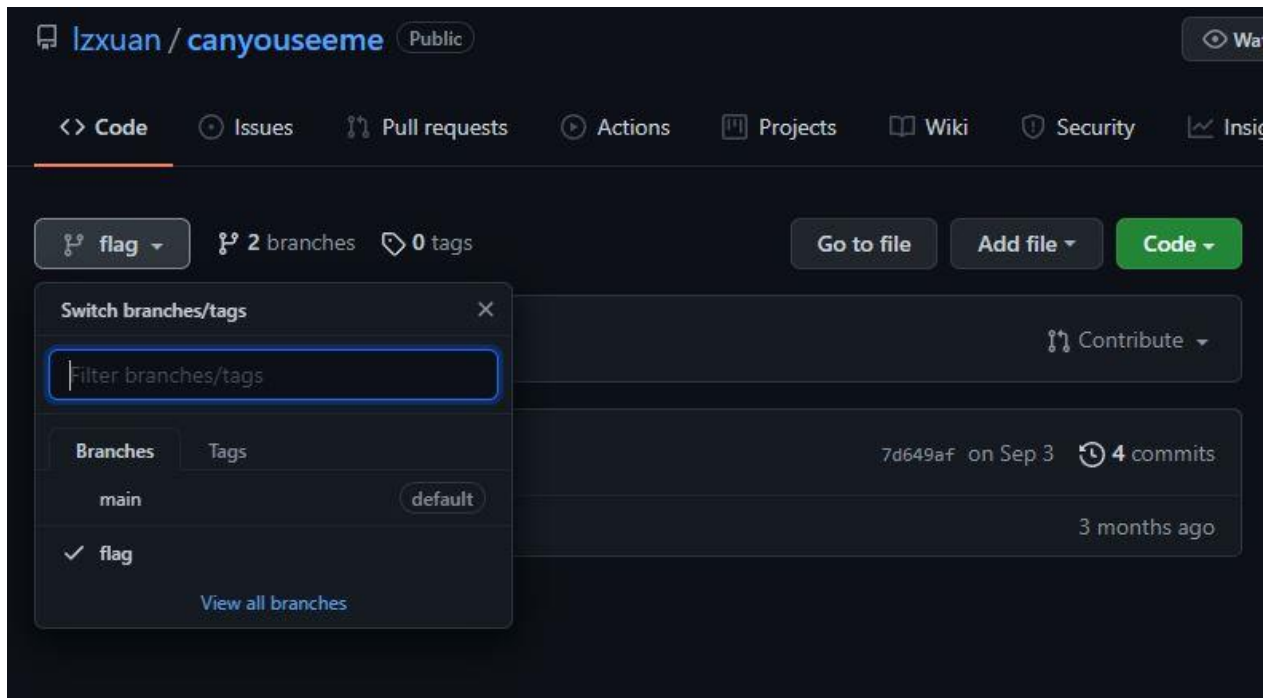


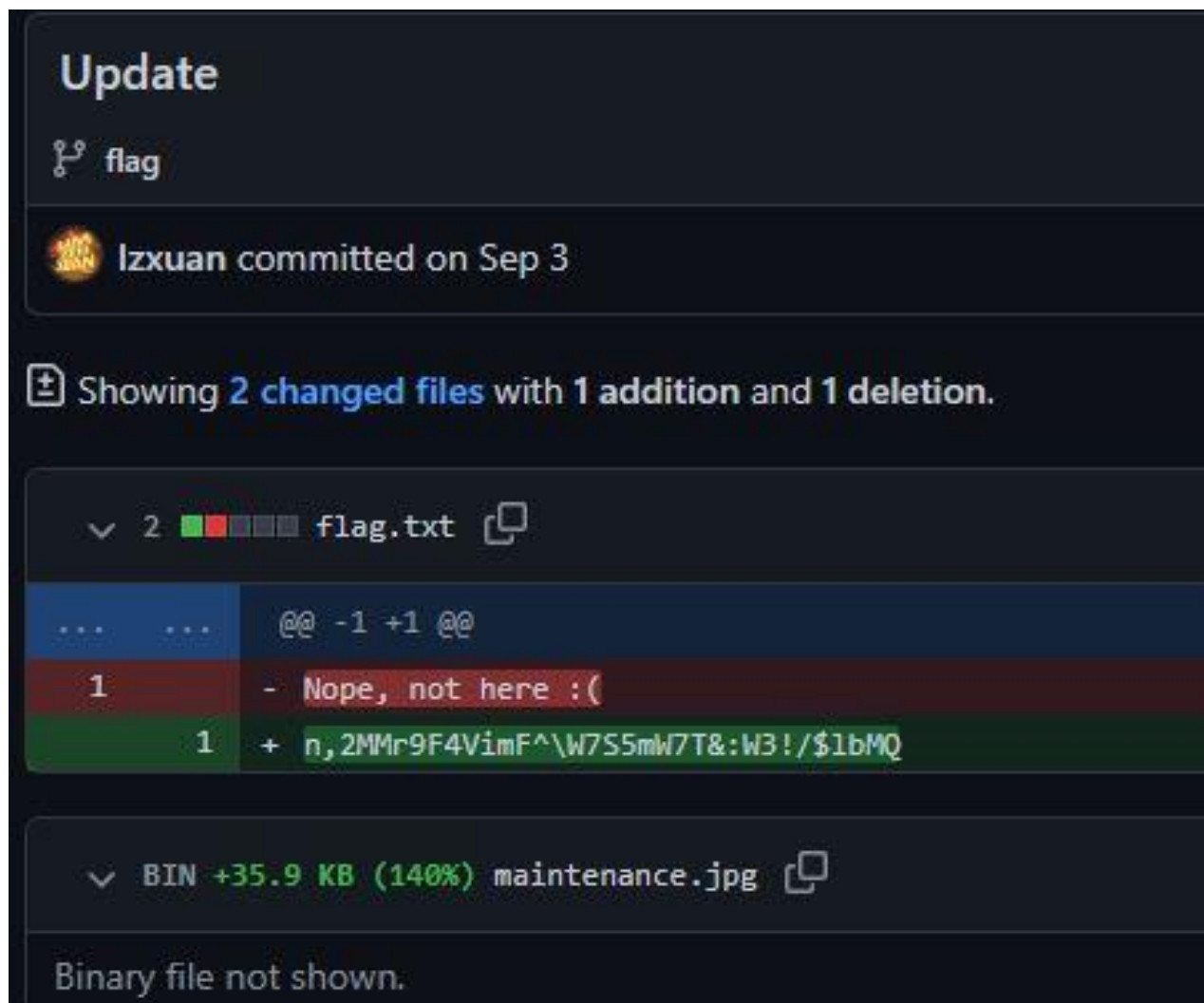
Flag: **verdesec{beginner_re_challenge}**

Flag: **verdesec{nonstop_netflag_november}**

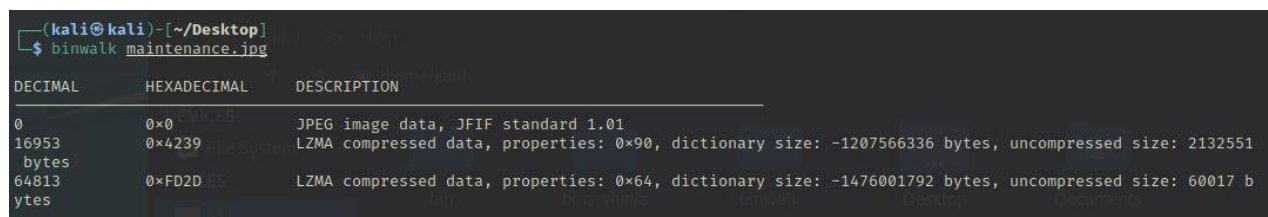
Flag #21 (1000)

The attempt to perform OSINT gathering on the challenge creators of TARUC CTF led us to the GitHub repository of the theme for this CTF `canyouseeme`. There are 2 forked branches, with the `flag` branch as our main interest, which contains 4 commits.





The “Update” commit has a 140% increase in terms of file size, which indicates some kind of data alteration or addition.



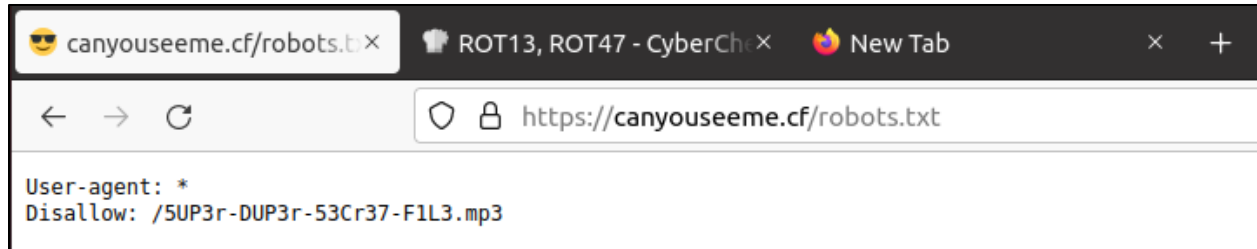
After verifying with Binwalk, the modified .JPG file indeed contains embedded data. The compressed data could not be extracted with the -e flag. Therefore, we moved on to visual steganography using Stegsolve.



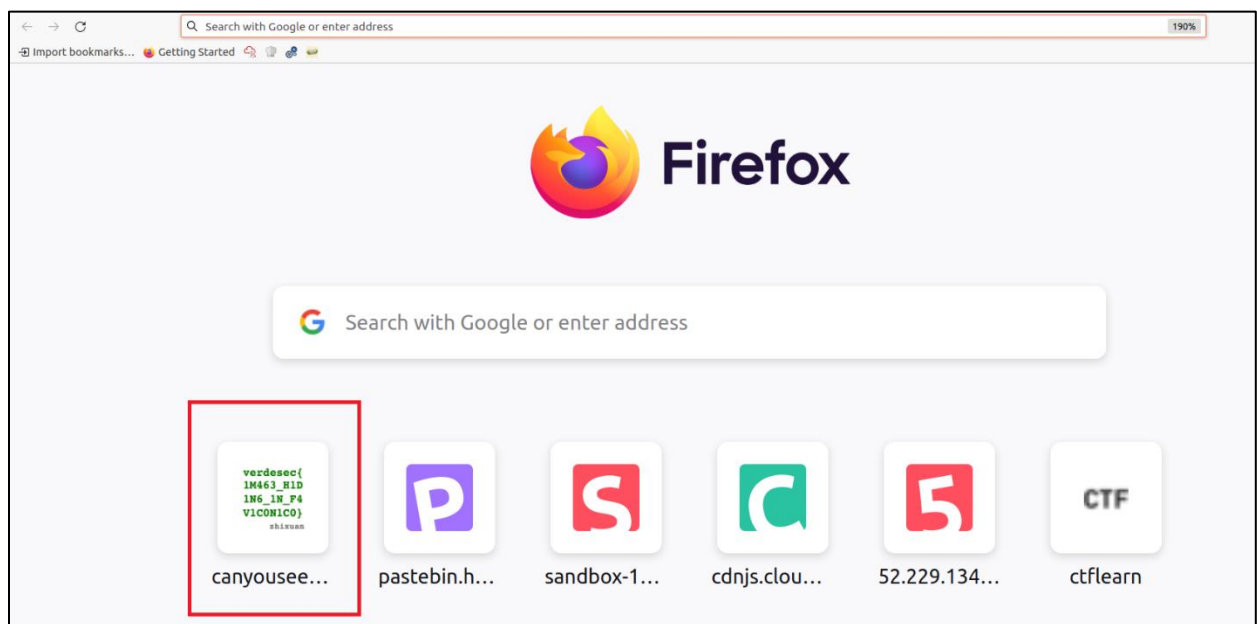
Flag: **verdesec{3Y3516H7_73571N6_ON3_7W0_7Hr33}**

Flag #23 (992)

This process of finding this flag is associated with the user-agent whitelist (robots.txt) challenge. However, we did not manage to solve the robots challenge.



Luckily, our best browser in the world pinned the robots.txt directory into one of the quick access slots. If we look closer, a hidden flag is revealed in the quick access graphic, which turns out to be a valid flag for the favicon challenge.



Flag: verdesec{1M463_H1D1N6_1N_F4V1C0N1C0}