Decoding a Cobalt Strike Downloader Script With CyberChef

By Matthew

Published: 2024-08-04 · Archived: 2026-04-05 19:41:11 UTC

Introduction

We recently encountered a short .HTA script on Malware Bazaar that was linked to the Cobalt Strike toolkit.

The script utilises basic obfuscation that can be removed using CyberChef and a text editor. This blog will cover our decoding process, including how to decode the following obfuscation methods

- Base64
- URL Encoding
- Excessive Spacing

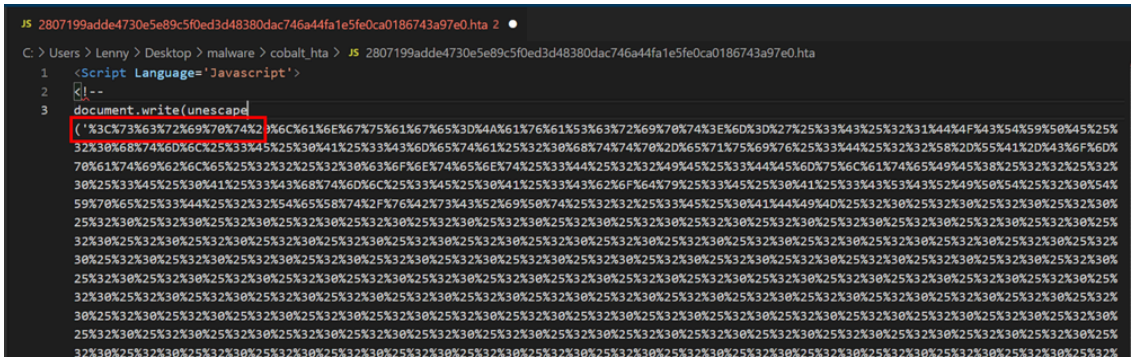
Original File

The file used for this analysis can be found on Malware Bazaar at the following link.

`2807199adde4730e5e89c5f0ed3d48380dac746a44fa1e5fe0ca0186743a97e0`

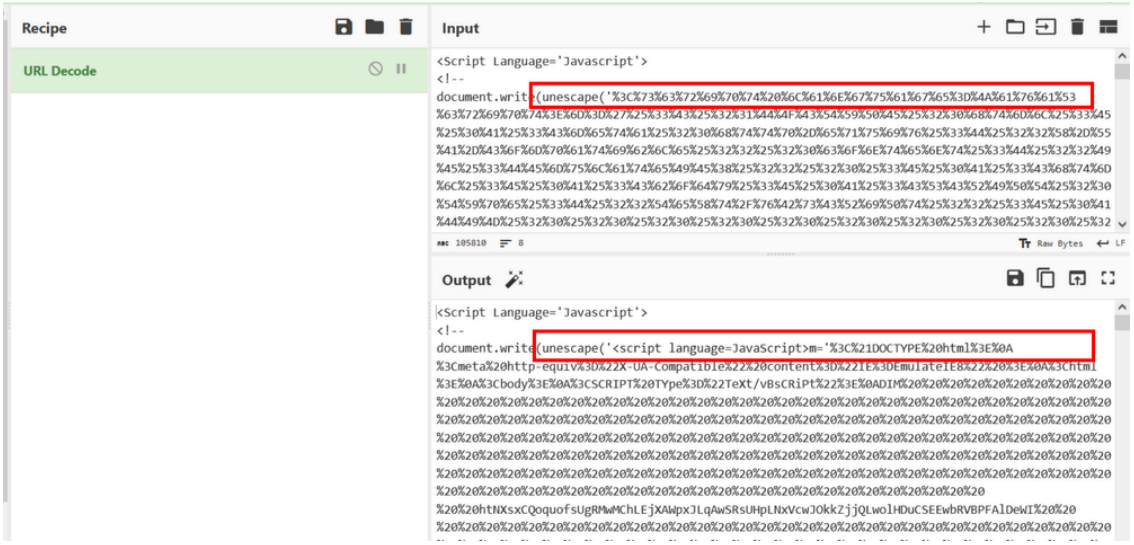
Analysis

The .HTA file in it's initial state contains a small amount of HTML followed by a large batch of URL encoded characters.



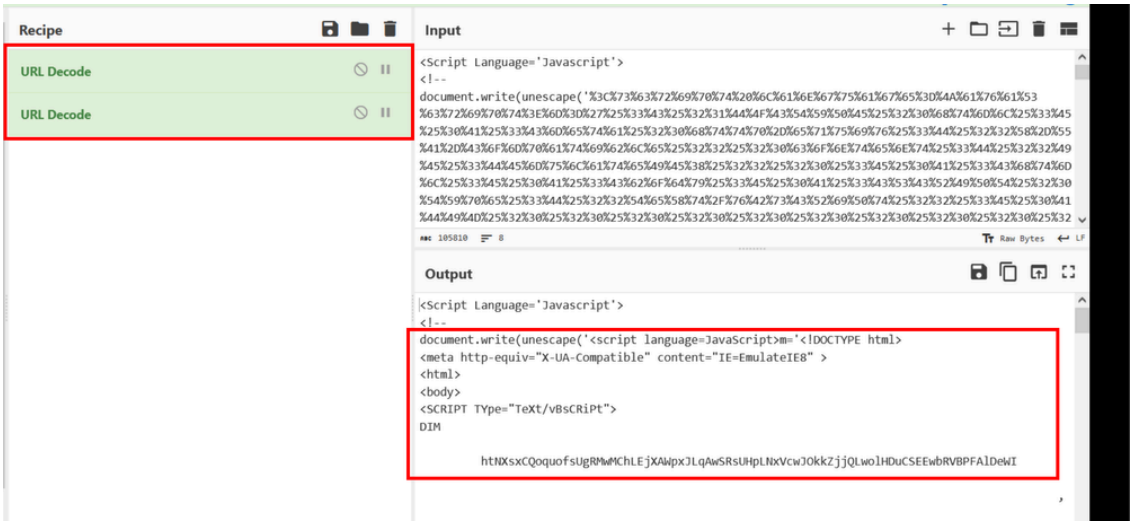
The first step to decoding the URL encoded content is to use a tool like CyberChef and the `URL Decode` operation.

As can be seen below, this decodes the content but reveals another layer of URL encoding.



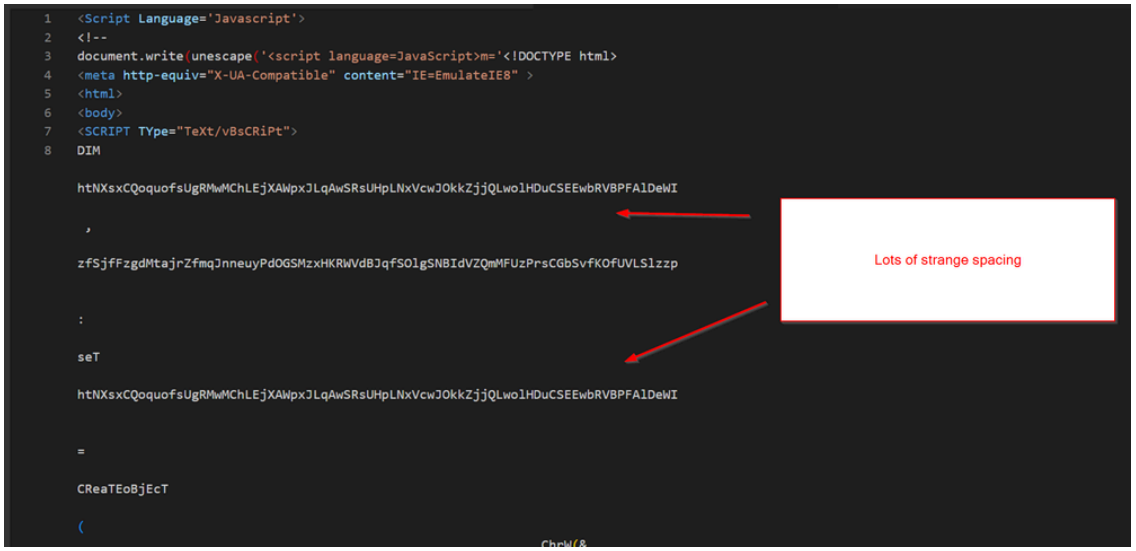
The second layer of URL encoding can be resolved with another URL Decode operation.

Applying the filter again removes the last of the URL encoding.



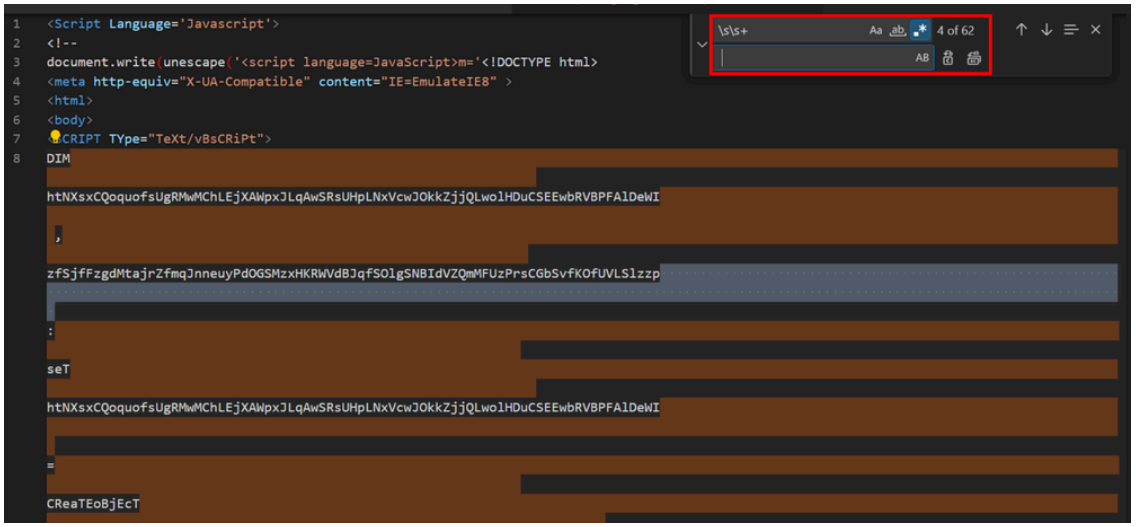
The content can now be moved back to a text editor for additional analysis.

Although the script is removed of URL encoding, the script now employs blobs of spaces to hinder analysis. This can be seen in the screenshot below.



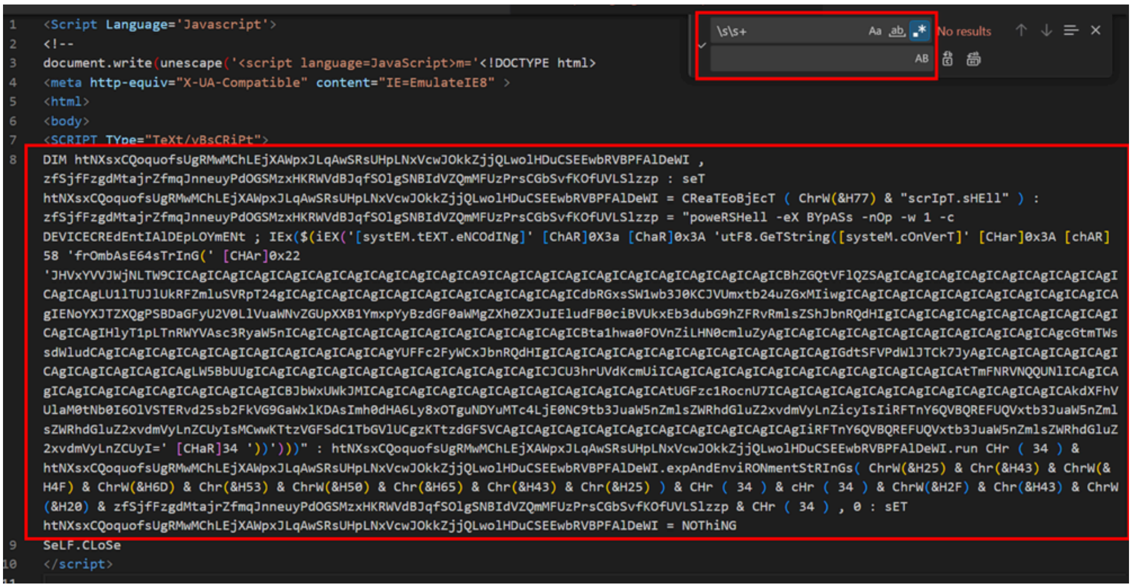
The spacing can be removed manually by highlighting and deleting, but a more efficient means is to use a regular expression to remove occurrences of two or more whitespace characters `\s`

By performing a search and replace with the `\s\s+` query, we can see the excessive spacing is highlighted and matched correctly.



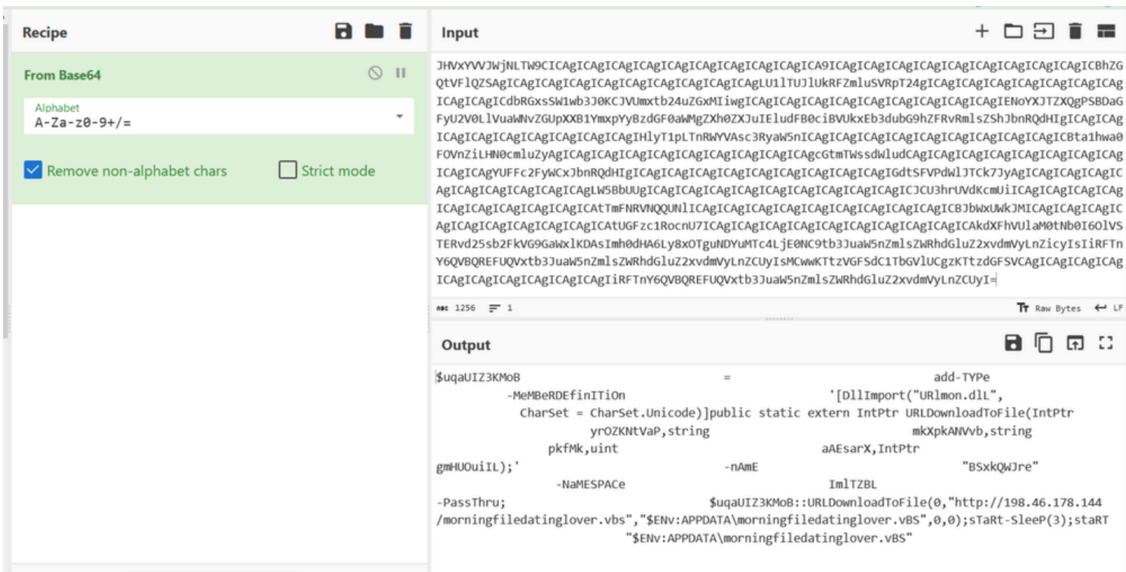
By specifying a replace value of a single space, the content can be cleaned up significantly.

The script content can now fit easily into a single screenshot.



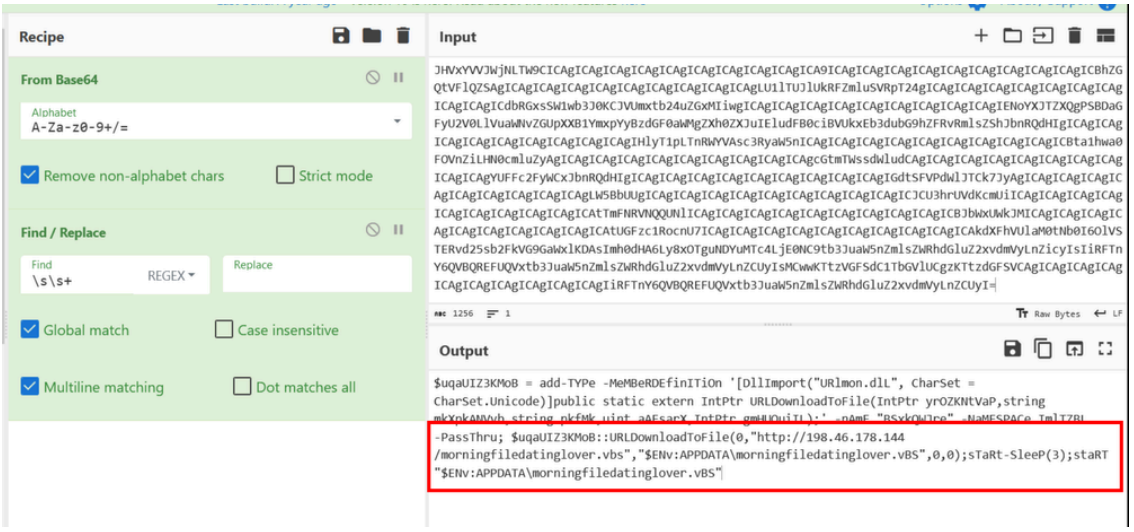
A large blob of base64 content can be seen in the resulting content. This is alongside the key giveaway of `frombase64string` with random upper and lower casing.

The base64 blob can be copied and decoded in CyberChef using `From Base64`



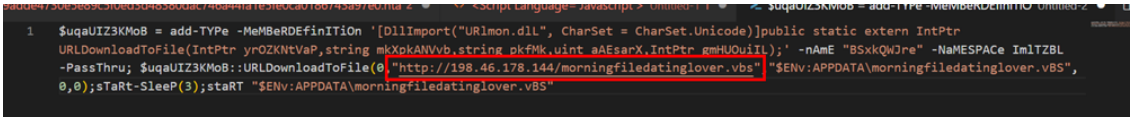
The resulting content contains more excessive spacing. The same regular expression technique as before can be re-applied to fix this.

Below we can see the From Base64 operation and the removal of excessive spacing via regex.



After the spacing is removed, it becomes clear that the code is a downloader utilising the `URLDownloadToFile` function.

The address of the next stage file is also clearly visible, and contains the value `http://198.46[.]178.144/morningfiledatinglover[.]vbs`



Source: https://www.embeeresearch.io/decoding-a-cobalt-strike-downloader-script-with-cyberchef/