Infection flow analysis of an agent Tesla delivered through a phishing email with a .js linked through the contained URL.

The content of the javascript downloaded from the web contains a bunch of obfuscated variables but it does not appear to be used in anything.

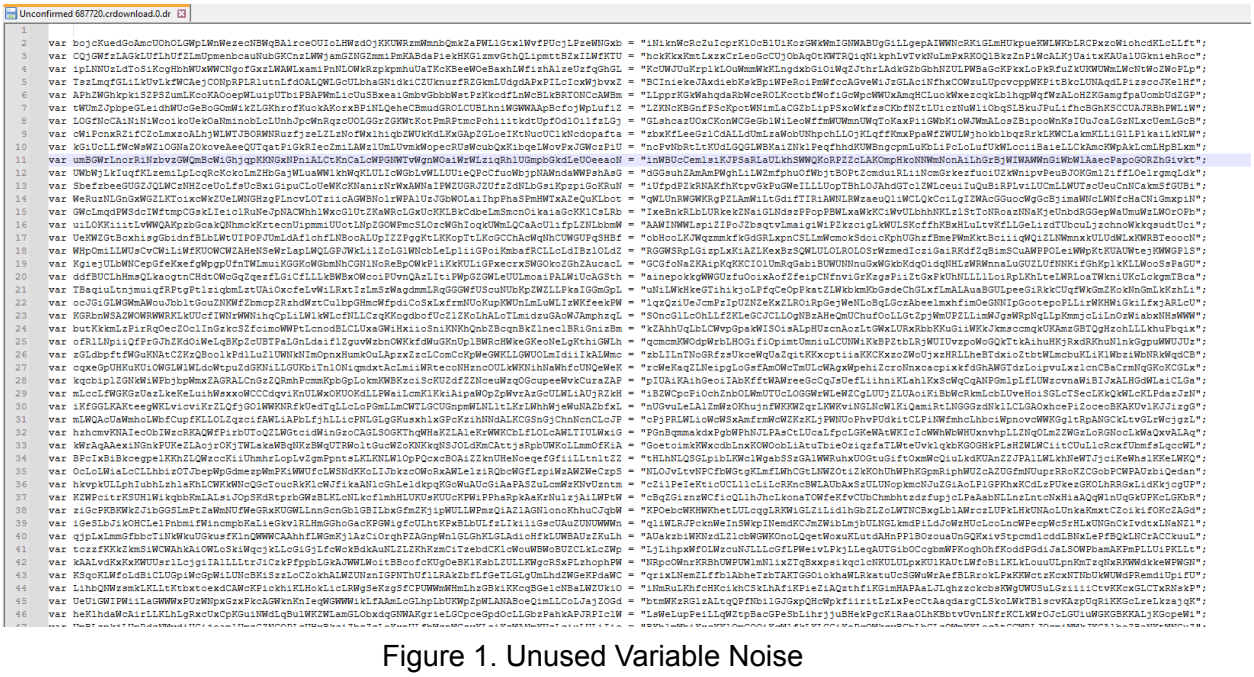

Figure 1. Unused Variable Noise

ActiveXObject is called and used to create a dictionary of some sort as well as launching powershell.

```
var lNCWCRCqiczeWnNmmWcdKKUULWKkHkdWePiqeLNfKgbUcxcGqtxLelRZLlbURGcL(
 var betle = new ActiveXObject("Scripting.Dictionary");
    var codonophone = carcaju(betle);

    if (codonophone === "lustrino") {
        switch (betle.Item("action")) {
            case metatarso:
                codonophone = rondear(betle);
                break;
            case invencionar:
                codonophone = folchlore(betle);
                break;
            case enojadamente:
                codonophone = rechan(betle);
                break;
            case exsolver:
                codonophone = paixoeiro(betle);
                break;
            case suinophobia:
                codonophone = rondear(betle);
                break;
            default:
                Usage(true);
```

Figure 2. Dictionary

Figure 3. WScript.Shell

The remainder of the script on this javascript file goes back to a repeat of all the unused variables with seemingly random values just to add noise.

The opened powershell command runs an obfuscated script that can be decoded with cyberchef through the use of both fromBase64 and remove Null. The following result is shown below.

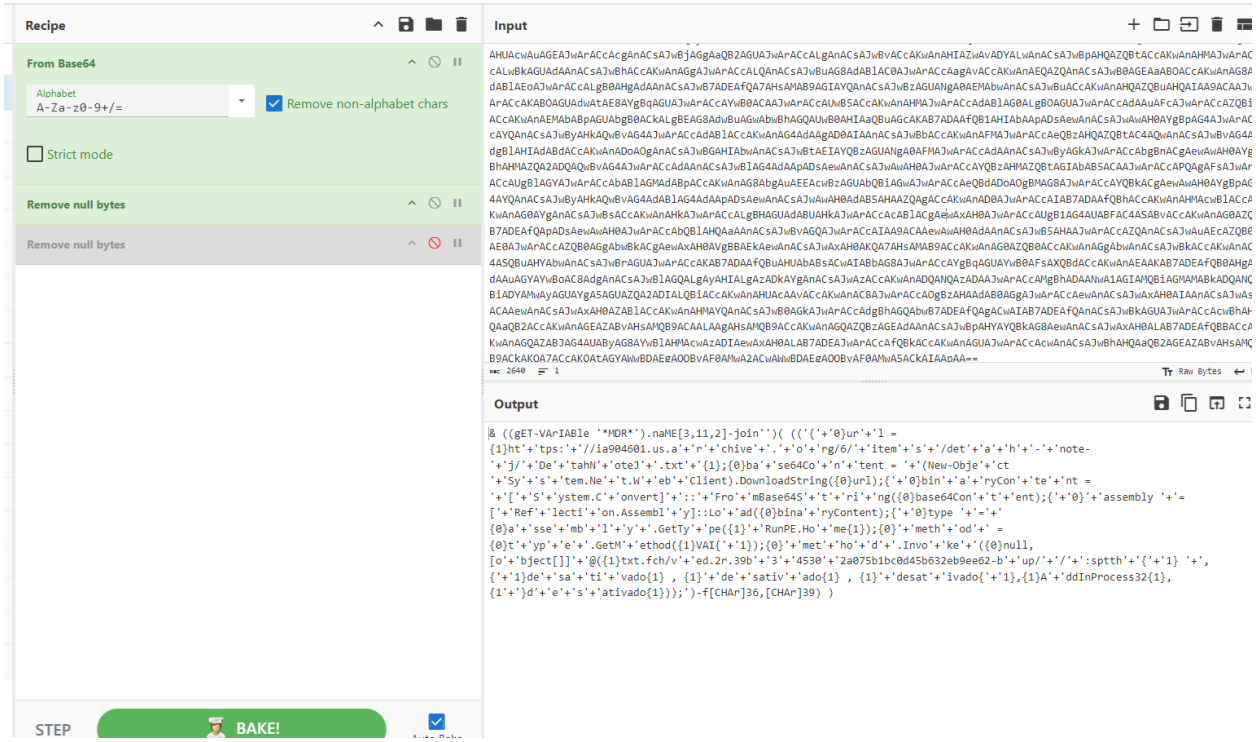

Figure 4. Cyberchef Decoding

Since the decoded string still runs the command by appending the string values together, I've put it into a string replacer to remove it and used the final output to access the final ".txt" file that is loaded into the assembly.

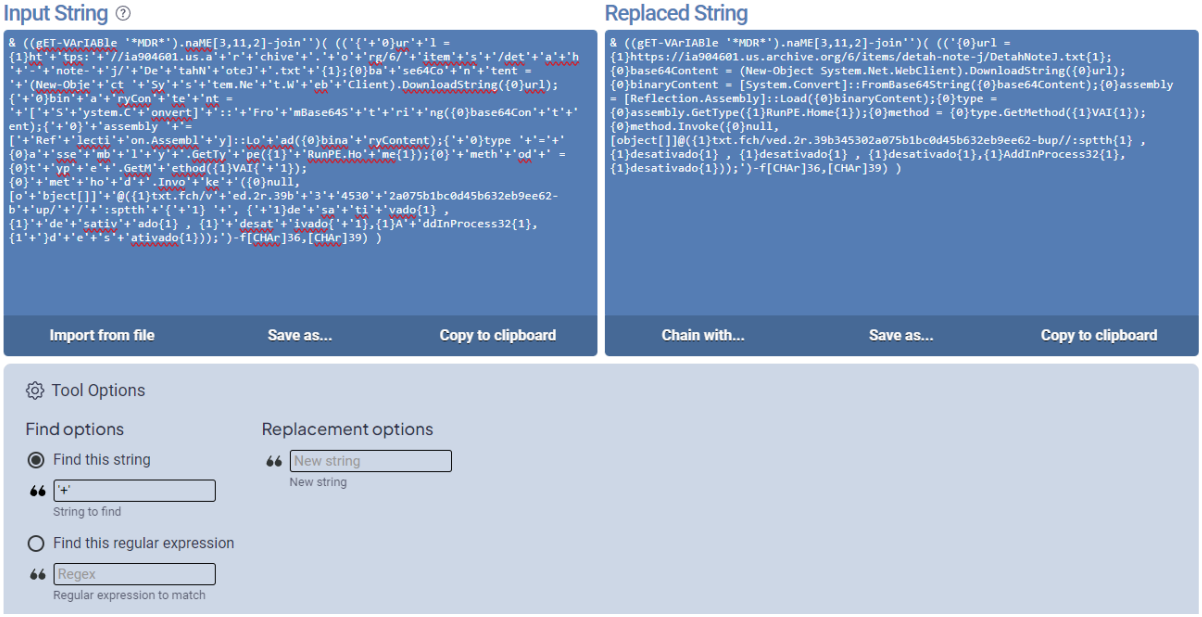

Figure 5. Concatenating the Strings Together

The contents of the encoded txt file which will be compiled into the assembly are shown below.

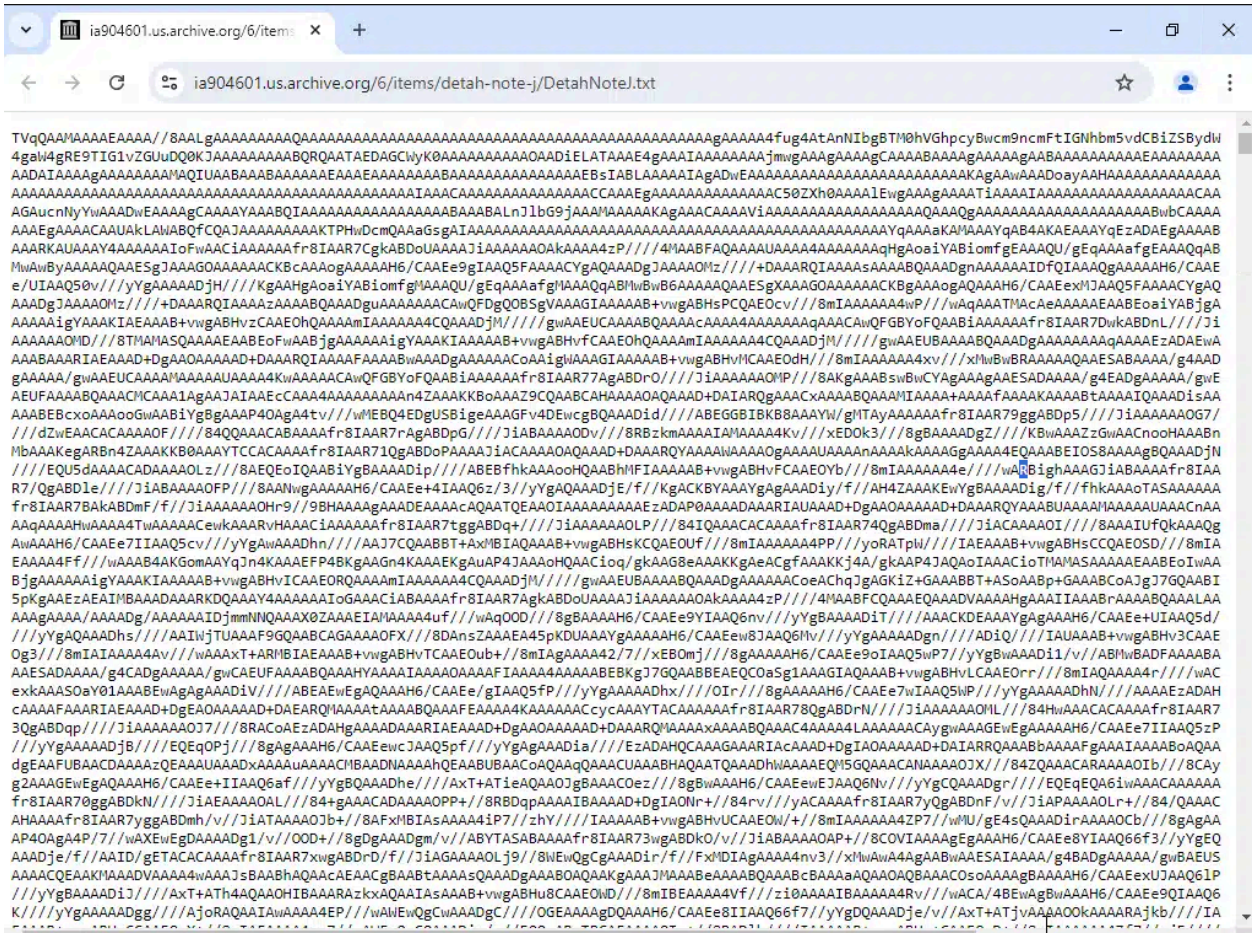

Figure 6. Content of .txt File.

Running the text file into cyberchef to decode the binary shows that it has an MZ header and is likely going to be the Agent Tesla payload.

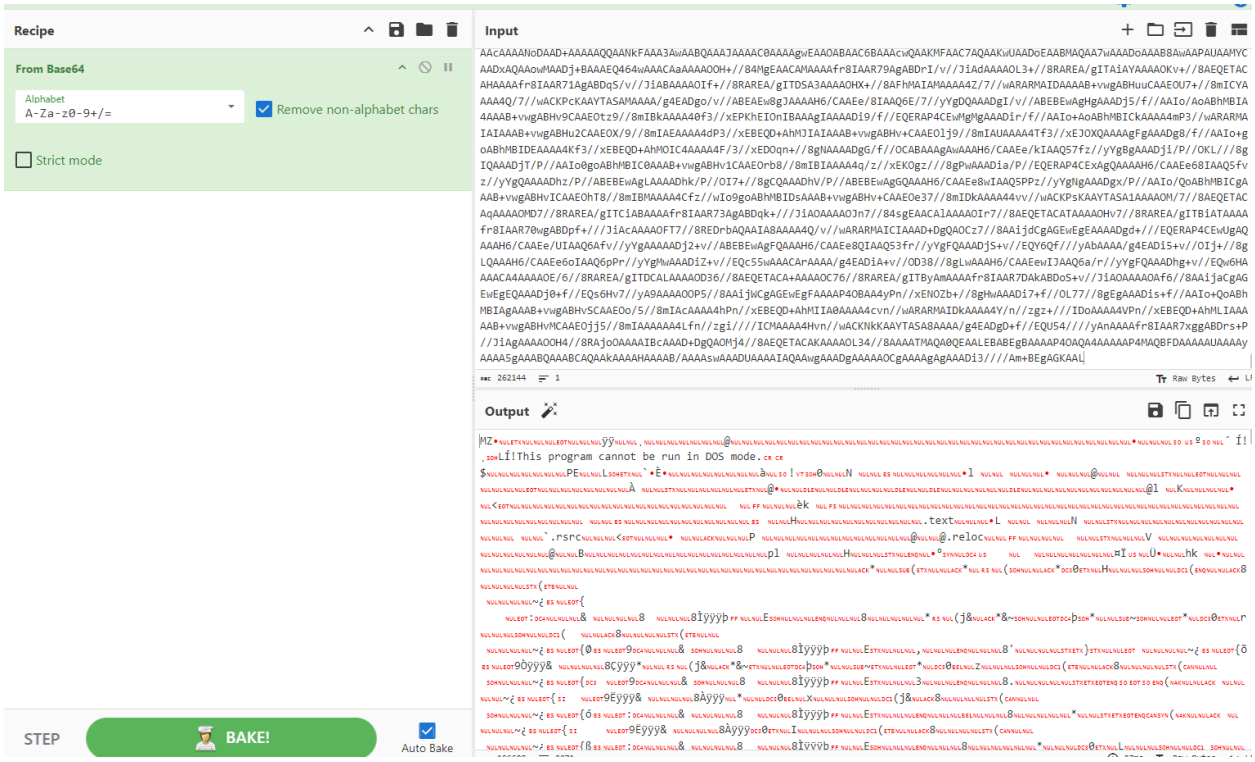

Figure 7. Compiled txt File Decoded