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

Analizë teknike për skedarin keqdashës
Display_10.exe

Versioni: 1.0
Datë: 25/06/2025

PËRMBAJTJA

Informacione Teknike 4

Analiza e skedarit Display_10.exe 4

Analiza e skedarit rawio.sys 11

Indikatorët e Kompromet imit 13

Rekomandime 13

Figura 1 Gjuha e kompiluesit dhe certifikata 4
Figura 2 Funksioni FUN_1400294c0 5
Figura 3 Evidentimi i rawio.sys në direktorinë temp 5
Figura 4 Krijimi i skedarit rawio.sys 6
Figura 5 Prapashtesat e skedarëve 6
Figura 6 Kërkimi i disqeve 7
Figura 7 Leximi i skedarëve dhe direktorive 8
Figura 8 Krijimi i servisit keqdashës RAW_IO 8
Figura 9 sectorio.sys 9
Figura 10 Serviset e symantec etj 10
Figura 11 Certifikata e rawio.sys 11
Figura 12 servisi I rawio.sys 11
Figura 13 source kodi i sectorio 12
Figura 14 Pas ekzekutimit të display_10.exe 13

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

Është evidentuar prezenca e një skedari keqdashës në sistemet e IT të Bashkisë Tiranë, ku është kryer shpërndarja dhe ekzekutimi në sistemet e operimit të serverave Windows dhe pajisjeve fundore Windows. Skedari ka rol shkatërrues dhe pasi ekzekutohet prish sistemin e operimit, mbishkruan skedaret e tipit dll,exe,docx etj duke i shkatërruar origjinalitetin.

Analiza e skedarit Display_10.exe

Skedari **display_10.exe** është një skedar i tipit i *ekzekutueshëm* i shkruajtur në gjuhën C/C++, i cili përmban një certifikatë legjitime, pra është i nënshkruar dhe verifikuar zyrtarisht me emrin **Tengku Zamzam.** Falë kësaj, ky skedar bëhet edhe më i besueshëm si një skedar legjitim.

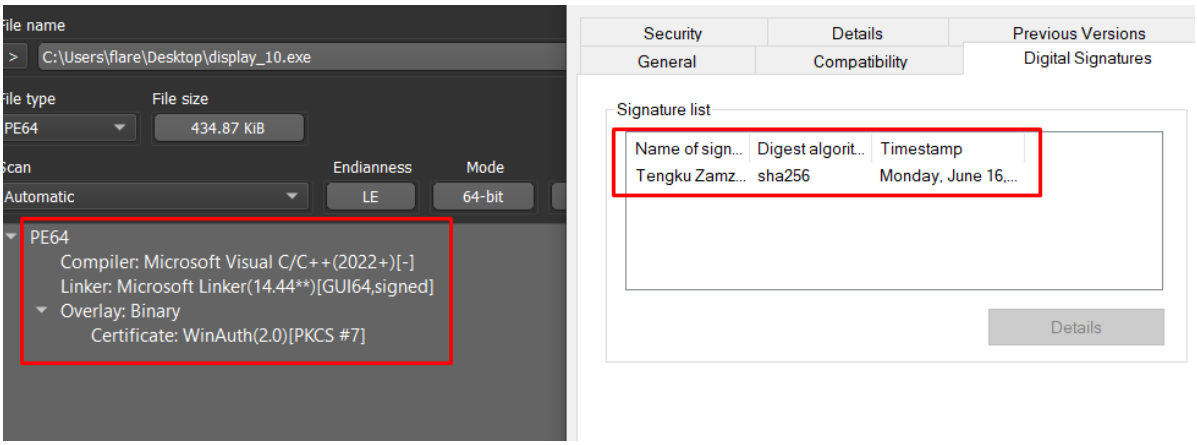

Figura 1 Gjuha e kompiluesit dhe certifikata

Gjatë analizës së kodit u evidentuan disa funksionalitete të këtij skedari keqdashës.

FUN_1400294c0 është një funksioni i cili:

- Kontrollon privilegjet e procesit aktual (si **SeShutdownPrivilege**).
- Krijon një file në disk në një vendndodhje të caktuar në desktop.
- Përgatit një listë skedarësh me emra në dukje si module DLL, EXE, sys, piz, gif, rar, z7, tar etj.
- I përpunon këto skedarë me funksione të tjera të brendshme si FUN_14002ac80, FUN_140027e58, FUN_14002a670, FUN_14002b554.
- Ekzekuton një **kontroll të pajisjeve/diskut**, dhe evidentohet vargu i karaktereve **C:\Windows\System32\drivers\beep.sys**
- Në fund, përpiqet të **fitojë privilegjin për të mbyllur ose rindezur sistemin**, dhe nëse ia del, thërret **ExitWindowsEx**.

Figura 2 Funksioni FUN_1400294c0

Funksioni **FUN_140025cc8** merr path-in e skedarit të përkohshëm të sistemit (Temp) nëpermjet **GetTempPathA(0x104, local_118);**

Ky funksion i Windows merr path-in e përkohshëm, p.sh.:
C:\Users\<username>\AppData\Local\Temp dhe ndërton një varg karakteresh **C:\Users\<username>\AppData\Local\Temp\\rawio.sys**
Kjo na jep idenë që kemi të bëjmë me një skedar të llojit *driver* në një direktori jo të zakonshme.

```
undefined (*) [32] FUN_140025cc8(undefined8 param_1,undefined (*param_2

{
  CHAR local_118 [272];

  FUN_14004bcc0(param_2,0,0x104);
  GetTempPathA(0x104,local_118);
  strncpy((char *)param_2,local_118,0x104);
  strncat((char *)param_2,"\\",0x104);
  strncat((char *)param_2,"rawio.sys",0x104);
  return param_2;
}
```

Figura 3 Evidentimi i rawio.sys në direktorinë temp

Nëse rikthehemi në funksionin e mëparshëm **FUN_1400294c0,** pikërisht këtu ndodh krijimi i këtij skedari rawio.sys .

```
pvVar10 = CreateFileA(local_158,
            0x10000000,      // GENERIC_READ | GENERIC_WRITE
            0,
            NULL,
```

2, // CREATE_ALWAYS
0x80, // FILE_ATTRIBUTE_NORMAL
NULL);

Krijohet rawio.sys me të drejtë shkrimi/leje për lexim-shkrim.
Nëse ekziston më parë, **fshihet dhe krijohet nga e para.**

Në pjesën e kodit si më poshtë shkruan në skedar **0x2D88 byte** nga **bufferi &DAT_140065390.**
Kjo është përmbajtja e driver-it rawio.sys që po krijohet në disk.

```
if ((pvVar10 == (HANDLE)0xffffffffffffffff) ||
   (WriteFile(pvVar10,&DAT_140065390,0x2d88,(LPDWORD)&local_5f8,(LPOVERLAPPED)0x0),
   (int)local_5f8 != 0x2d88)) {
  return 0xffffffff;
}
CloseHandle(pvVar10);
```

Figura 4 Krijimi i skedarit rawio.sys

Më poshtë në këtë funksion emrat dhe prapashtesat janë të ruajtura si vlera *hexadecimal*. Po, kjo pjesë e kodit, tregon një grup emrash të skedarëve me prapshtesa të zakonshme si **.backup, .config, .gif, .pst, .ost, .bak,exe,dll etj.** Të gjitha këto zakonisht janë objektiva të ransomware apo programeve të tjera keqdashëse.
Më pas përgatitet për të përsëritur mbi strukturën që përmban këto emra.
Kjo do përdoret më vonë për të:

- Skanuar skedarët në disk
- Kërkuar në disqe lokale për skedar që përfundojnë me këto prapashtesa.

```
Decompile: FUN_1400294c0 - (display_10.exe)
_local_598 = ZEXT716(0x6769666e6f632e);
local_568 = 4;
uStack_564 = 0;
uStack_560 = 0xf;
uStack_55c = 0;
auStack_573 = SUB1611(ZEXT816(0),5);
local_578 = (undefined  [5])0x7473702e;
local_548._8_4_ = 0xf;
local_548._0_8_ = 4;
uStack_53c = 0;
stack0xfffffffffffffaad = SUB1611(ZEXT816(0),5);
local_558._0_5_ = 0x74736f2e;
local_528 = 4;
uStack_524 = 0;
uStack_520 = 0xf;
uStack_51c = 0;
local_538._5_11_ = SUB1611(ZEXT816(0),5);
local_538._0_5_ = 0x6b61622e;
stack0xfffffffffffff9d0 = (LUID)local_518;
_local_638 = (undefined (*) [32])local_5b8;
FUN_14002ac80((undefined (**) [16])local_338,(undefined (**) [32])local_638);
ppvVar15 = (LPVOID *)local_518;
lVar20 = 5;
do {
```

Figura 5 Prapashtesat e skedarëve

Më poshtë jepet një pjesë kodi në mënyrë manuale për të **ndërtuar një path në wchar_t (Unicode string)**.

```
local_268._0_1_ = 'C';       // fillimi i stringut
...
uStack_250._3_1_ = '\\';     // karakteret vazhdojnë
...
uStack_24c._3_1_ = 'p';      // "beep"
local_248 = 0x7379732e;      // ".sys" (me anë të vlerave hex)
```

Si përfundimi krijohet path wchar_t *path = **L"C:\\Windows\\System32\\drivers\\beep.sys"**

Më pas vijojmë të kërkojmë **drive-t logjikë në sistem**, si C:\, D:\, etj.
Për çdo drive të zbuluar, përdor disa struktura dhe funksione për të bërë **analizë ose përpunim të të dhënave në ato drive**.

```
    FUN_14002ab08((LPVOID *)local_5d8);
}
local_5e0 = 0;
while (uVar5 = local_5e0,
      local_5e0 < (ulonglong)(((longlong)local_3b8 - (longlong)local_3c0) / 0x18)) {
  ppauVar1 = local_3c0 + local_5e0 * 3;
  _local_5d8 = ZEXT816(0);
  local_5c8 = 0;
  DVar7 = GetLogicalDrives();
  local_647 = 0x41;
  while (auVar2 = _local_5d8, bVar4 = local_647, (char)local_647 < '[') {
    if ((DVar7 >> ((int)(char)local_647 - 0x41U & 0x1f) & 1) != 0) {
      _local_638 = (_TOKEN_PRIVILEGES)ZEXT816(0);
      local_628 = 0;
      local_620 = 0;
      FUN_14002ad80((undefined (*) [32])local_638,local_647,1);
      pauVar9 = FUN_140028040((undefined (*) [32])local_638,(undefined (*) [32])&DAT_14005f2bc,2
      );
       local_618 = *(undefined (*) [16])*pauVar9;
```

Figura 6 Kërkimi i disqeve

Funksioni **FUN_14002917c**
Lexon çdo file dhe folder në një path të caktuar.
Nëse gjen ndonjë file që përputhet me një listë emrash (filtra), e **ruan atë në një listë**.
Nëse gjen një subfolder, futet thellë në të (rekursivisht).
I gjithë ky operacion ndodh me kujdes për **menaxhim manual të memories**, gjë që tregon se është pjesë e skedarit keqdashës.

```
  pauVar9 = *(undefined (**) [32])*param_1;
}
pauVar16 = param_3;
FUN_1400280b4((undefined (*) [32])local_1a8,param_2,param_3,pauVar9,
              *(ulonglong *)(*param_1 + 0x10),(undefined (*) [32])&DAT_14005f2b8,2);
lpFindFileData = (LPWIN32_FIND_DATAA)&local_168;
lpFileName = local_1a8;
if (0xf < local_190) {
  lpFileName = local_1a8[0];
}
hFindFile = FindFirstFileA((LPCSTR)lpFileName,lpFindFileData);
if (hFindFile != (HANDLE)0xffffffffffffffff) {
  do {
    if ((local_13c[0] != '.') ||
       ((local_13c[1] != '\0' && ((local_13c[1] != '.' || (local_13c[2] != '\0')))))) {
      if (*(ulonglong *)(*param_1 + 0x10) == 0x7fffffffffffffff) {
```

Figura 7 Leximi i skedarëve dhe direktorive

Funksioni **FUN_140025d40**

Ky funksion:

1. **Hap Service Control Manager** me OpenSCManagerA(NULL, NULL, 2) (akses për krijim dhe modifikim të servisi).
2. **Krijon një service të ri të quajtur RAW_IO** me **CreateServiceA**(...), dhe si path të executable-it përdor param_1 (parametër që vjen nga jashtë).
   RAW_IO është emri i servisit **(i dyshimtë)**.
   param_1 është path i executable-it ose driver-it që do të ngarkohet si service.
3. **Starton servicin** me StartServiceA(...).
   - Nëse startimi nuk ka sukses por GetLastError() == 0x420 (ERROR_SERVICE_ALREADY_RUNNING), e konsideron gjithsesi sukses.
4. Nëse dështojnë disa hapa, e **kontrollon nëse RAW_IO është krijuar më parë** me OpenServiceA(...).
5. Në fund, kthen 1 nëse suksesshëm, 0 nëse jo.

param_1 është rruga e skedarit që do të përdoret si executable për servisin.
Pra, në rastin konkret driver **rawio .sys** n do të jetë driveri rawio.sys I krijuar në direktorinë temp.

```
Decompile: FUN_140025d40 - (display_10.exe)
{
  BOOL BVar1;
  DWORD DVar2;
  SC_HANDLE hSCManager;
  SC_HANDLE pSVar3;
  undefined *puVar4;
  FILE *pFVar5;
  undefined8 uVar6;

  hSCManager = OpenSCManagerA((LPCSTR)0x0,(LPCSTR)0x0,2);
  uVar6 = 0;
  if (hSCManager != (SC_HANDLE)0x0) {
    pSVar3 = CreateServiceA(hSCManager,"RAW_IO","RAW_IO",0x10030,1,3,1,param_1,(LPCSTR)0x0,
                            (LPDWORD)0x0,(LPCSTR)0x0,(LPCSTR)0x0,(LPCSTR)0x0);
    if (pSVar3 != (SC_HANDLE)0x0) {
      BVar1 = StartServiceA(pSVar3,0,(LPCSTR *)0x0);
      if ((BVar1 != 0) || (DVar2 = GetLastError(), DVar2 == 0x420)) {
        uVar6 = 1;
      }
      CloseServiceHandle(pSVar3);
      CloseServiceHandle(hSCManager);
      return uVar6;
    }
    puVar4 = FUN_140035634(1);
```

Figura 8 Krijimi i servisit keqdashës RAW_IO

Funksioni **FUN_14002b620** është funksioni më kritik i analizës. Ai përdor një teknikë për **shkrim të drejtpërdrejtë në disk**, përmes një **driver-i të quajtur sectorio.sys**, për të **modifikuar sektorë të diskut në nivel të ulët**, për të manipuluar boot sector ose mbishkruar skedarë .exe/.dll si pjesë e një teknikë "wiper" pra në shkatërrimin e të dhënave.
Ky funksion **përdor këtë driver (sectorio.sys)** për të kryer një **IOCTL (Input/Output Control)** që:

- **Mundëson shkrimin direkt në sektorët e diskut** (shumë e rrezikshme).
- **Kalon mbrojtjet e Windows** për shkrime në MBR/**Volume** Boot Record ose direkt në skedarë të mbrojtur.
- **Bypass-on endpoint protection**, duke shmangur shkrimin përmes API-ve të Windows-it.

Figura 9 sectorio.sys

Gjatë analizës u evidentua dhe një pjesë kodi në një funksion
pa emër ku janë të koduara një sërë **emrash të skedarëve** si **LuCallbackProxy.exe ccScHvScht.exe SmSc.exe (nga 0x6578652e636d53) AccApp.exe SysmSrv.exe SysmEAFE.exe IR.exe SysclmScvs**
Këto janë të gjitha emra të lidhur me **Symantec/Norton/Security Suite exe**.

Kjo pjesë kodi është projektuar për të anashkaluar mbrojtjen e **Symantec** duke përdorur emra të proceseve legjitime të tij, gjë që mund të ndalojë inicializimin e moduleve mbrojtës (për shembull, nëse Symantec përdor kontroll me emra procesesh për self-protection).

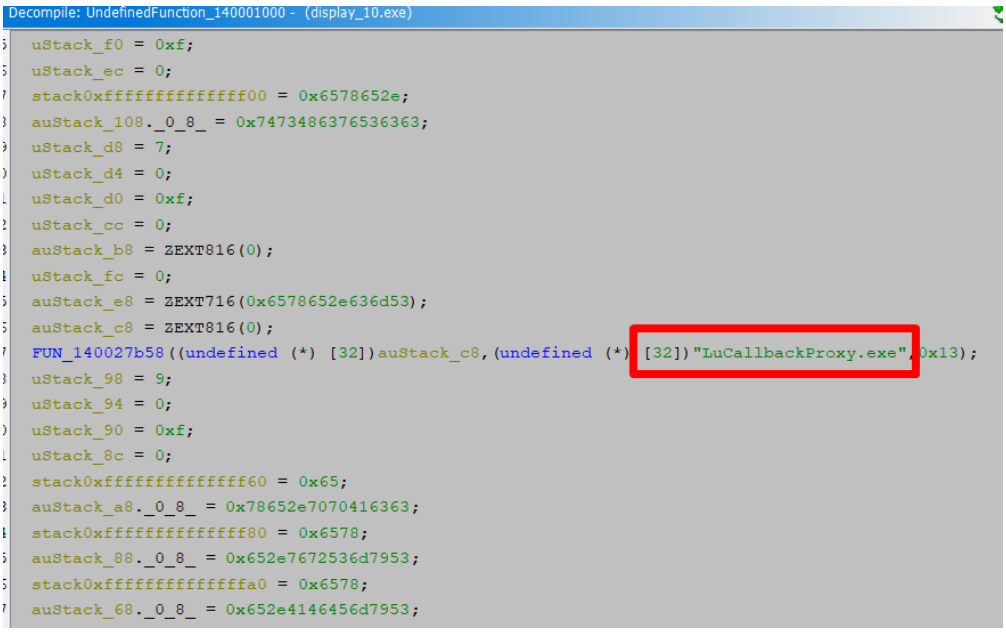

Figura 10 Serviset e symantec etj

Funksioni **FUN_14002b390 ë**shtë funksioni që përdoret nga skedari keqdashës për të komunikuar me driver-in e vet sectorio.sys dhe për të ekzekutuar **operacione RAW në disk** (lexim/shkrim drejtpërsëdrejti në sektorë).

- **Zgjedh volumin** në bazë të driveIndex.
- **Verifikon** që volumi është **NTFS** (shumica e sistemeve Windows).
- **Hap diskun** në **RAW I/O** (anashkalon sistemin e skedarëve dhe shumicën e mbrojtjeve AV).
- **Paketon** madhësinë/offset-in (local_e0) dhe e dërgon te driver-i përmes IOCTL-it të personalizuar.
- **Driver-i sectorio.sys** ekzekuton shkrimin/leximin fizik dhe kthen adresën reale të sektorit të prekur.
- **Përdor IOCTL jo-standard (0x560020)** → kërkon ekzistencën e sectorio.sys.
    - **Shkruan në disk në nivel sektori** → mund të fshijë, të dëmtojë ose të fshehë të dhëna pa u kapur nga AV-të.
- **Verifikon NTFS** për të siguruar përputhje me strukturën e disqeve Windows.
- **Mbështetet në path-et "\\.\X:"** – teknikë klasike e malwarëve/wiper-ave për të anashkaluar API-të e high level

Funksionaliteti kryesor i këtij skedari është shkatërrimi i skedarëve të sistemit operativ të windows dhe i skedarëve të vetë përdoruesve. Pra procesi që ndodh funksionon në formë të tillë: skedari legjitim i windows beep.sys përdoret si input për të mbishkruar skedarët nëpërmjet driverit rawio.sys që në fakt është sectorio.sys. Roli i driverit në këtë rast funksionon si një Proxy, nga ku i gjithë procesi i mbishkrimit nuk ndodh nga vetë skedari i ekzekutueshëm por nga vetë driveri. Skedari i cili përdoret si mbishkrim është skedari beep.sys i cili përdoret si parametër për të mbishkruar mbi skedarët e vetë sistemit operativ.

Analiza e skedarit rawio.sys

Skedari display_10.exe kërkon të drejta administratori për të funksionuar, pasi që të krijohet një servis i tipit kernel do të duhen të drejta administrative.Ajo cfare e bën këtë sulm të suksesshëm është certifikata legjitime që i është nënshkruar. Nuk bëhet fjalë për certifikatën e skedarit display_10.exe por për certifikatën e rawio.sys pasi ky është skedari i i cili përcakohet si binary executable në servisin.

Në një sistem modern Windows x64 me Secure Boot / DSE të aktivizuar, **sectorio.sys pa certifikate nuk mund të ngarkohet si driver kernel**, edhe pse mund të regjistrohet si shërbim. Do të ngarkohej vetëm nëse sulmuesi:

- çaktivizon **Driver Signature Enforcement** (Test Mode, Safe Mode me Disable integrity checks),
- përdor një metodë “bring-your-own-vulnerable-driver”,
- ose është në një sistem 32-bit / shumë i vjetër ku nënshkrimi nuk aplikohet

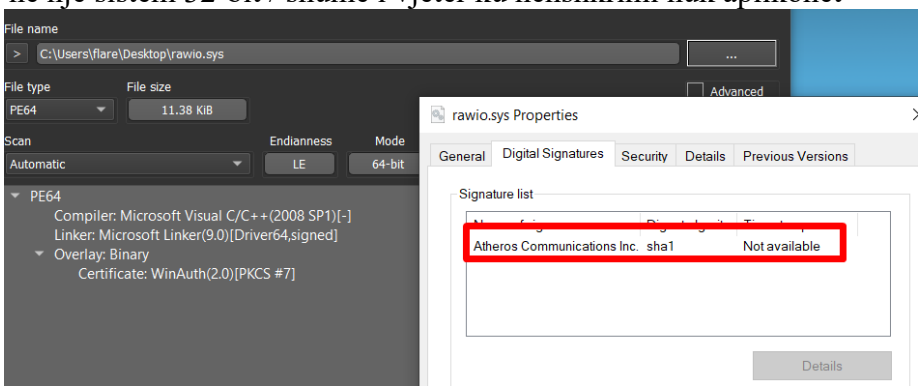

Figura 11 Certifikata e rawio.sys

Nëse bëjmë query mbi serviset e windows me emrin RAW_IO do evidentohet dhe pathi i rawio.sys i cili është në pathin e temp të përmendur dhe më pare.

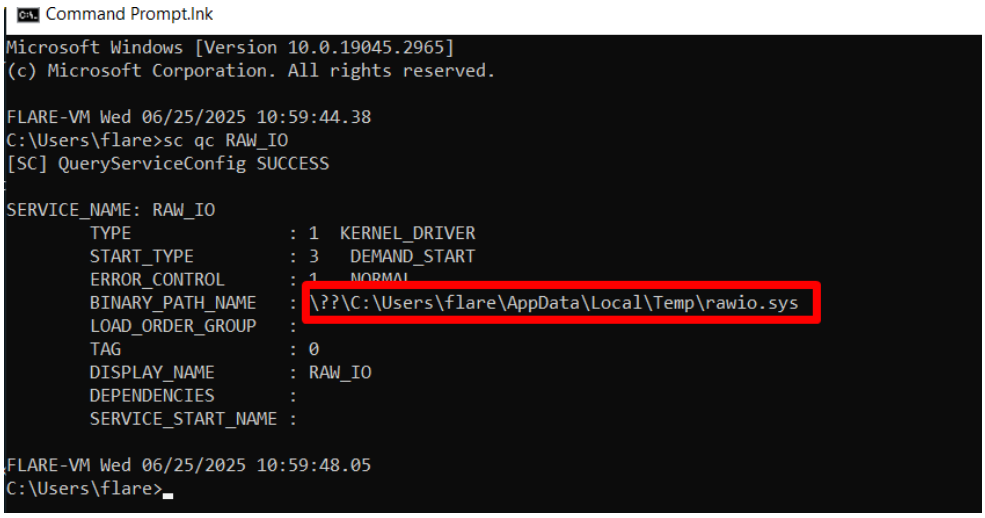

Figura 12 servisi I rawio.sys

Gjatë evidentimit dhe kërkimit të kërcënimeve u evidentua repository në github i cili u përdor për krijimin e këtij driveri **https://github.com/jschicht/SectorIo/**

Nga kodi i shkruar në gjuhë **C** u kuptua se ky **driver Windows x64 (kernel-mode)** ekspozon një objekt të quajtur **\\Device\\sectorio → \\DosDevices\\sectorio** dhe u lejon proceseve në "user-mode" të bëjnë **leximе/shkrime RAW** në sektorët e çdo disku ose ndarjeje (partition) që ekziston në sistem.

WCHAR g_szDeviceName[] = L"\\Device\\sectorio";
WCHAR g_szDosDeviceName[] = L"\\DosDevices\\sectorio";

IoCreateDevice(…, &szDeviceName, FILE_DEVICE_UNKNOWN, …, &gp_DevObj);
IoCreateSymbolicLink(&szDosDeviceName, &szDeviceName);
sectorio.sys ekspozon një pajisje **\\.\sectorio** që pranon një IOCTL (0x560020) për të lexuar/mbishkruar sektorë fizikë në çdo disk/partition NTFS. Driveri bën enumerate të **të gjitha objekteve të disk.sys** dhe ruan për secilin madhësinë e sektorit. Çdo proces user-mode që hap pajisjen mund të thërrasë IOCTL_SECTOR_WRITE dhe të kryejë **shkrim të drejtpërdrejtë në disk (RAW)** pa asnjë verifikim të privilegjeve, duke anashkaluar mekanizmat e integritetit të skedarëve dhe monitorimin e antiviruseve. Kjo e bën driverin një mjet ideal për **wiper, rootkit ose ransomware** që duan të manipulojnë diskun nën nivelin e sistemit të skedarëve.

Sectorio / sector.c

Code Blame 511 lines (406 loc) · 13.6 KB

```
20
21  #include "ntddk.h"
22  #include "ntdddisk.h"
23  #include "stdarg.h"
24  #include "stdio.h"
25  #include <ntddvol.h>
26
27  #include <mountdev.h>
28  #include "wmistr.h"
29  #include "wmidata.h"
30  #include "wmiguid.h"
31  #include "wmilib.h"
32  #include "sector.h"
33
34
35
36  WCHAR                       g_szDeviceName[] = L"\\Device\\sectorio";
37  WCHAR                       g_szDosDeviceName[] = L"\\DosDevices\\sectorio";
38  UNICODE_STRING          szDeviceName;
39  UNICODE_STRING          szDosDeviceName;
40
41  PDEVICE_OBJECT  gp_DevObj = NULL;
42
43  #pragma alloc_text(INIT, DriverEntry)
44
45  NTSTATUS GetGeometry(PDEVICE_OBJECT pDiskDevObj, PDISK_GEOMETRY pDiskGeo)
46  /*++
```

Figura 13 source kodi i sectorio

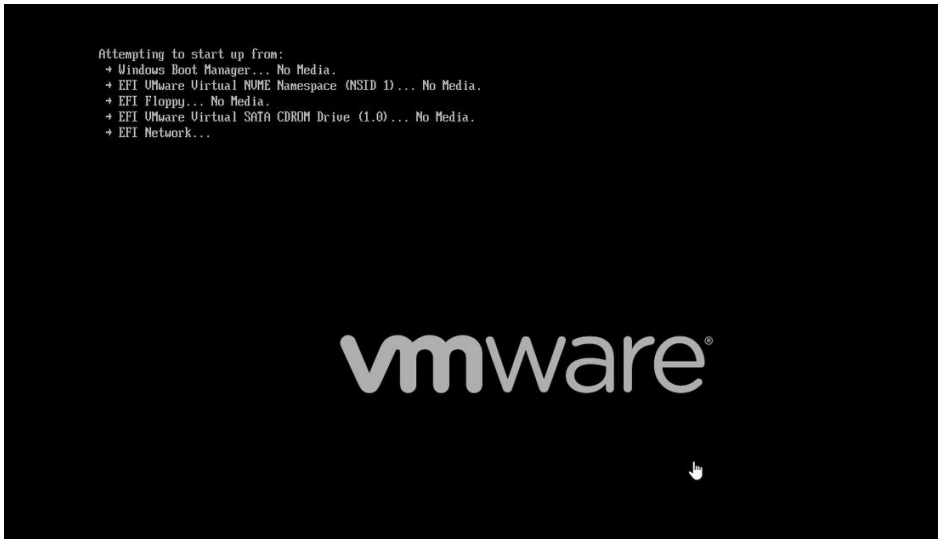

Figura 14 Pas ekzekutimit të display_10.exe

Indikatorët e Komprometimit

81EB22828306F3197B35FEF2035CEF2C548F587F8511902852964850023389D7	**Display_10.exe**
06B7A3CD3266449294EEEFB957965EA9F1354804696955E1EB1A7F9B515F5880	**Rawio.sys**

Rekomandime

Autoriteti Kombëtar për Sigurinë Kibernetike rekomandon:

- Bllokimin e menjëhershëm të Indikatorëve të Komprometimit, të përmendura më sipër në pajisjet tuaja mbrojtëse.
- Analizimin e vazhdueshëm të logeve që vijnë nga SIEM (Security information and Event Management).
- Trajnimin e 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).
- 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 menagjimin e fjalëkalimeve të Administratorëve Lokal.
- Aplikimin e filtrave të trafikut në rastin e aksesimit në distancë të hosteve (punonjësve/palë të treta/klientë).

- Implementimin e zgjidhjeve që kryejnë filtrimin, monitorimin dhe bllokimin e trafikut keqdashës ndërmjet aplikacioneve Web dhe internetit, Web Application Firewall (WAF).
- Kryerjen e analizave 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.
- Projektimin e zgjidhjeve për menaxhimin e aksesit të përdoruesve “Identity Access Management” për të kontrolluar identitetin dhe privilegjet e përdoruesve në kohë reale sipas parimit “zero-trust”.