Return of the kernel rootkit malware (on Windows 10)

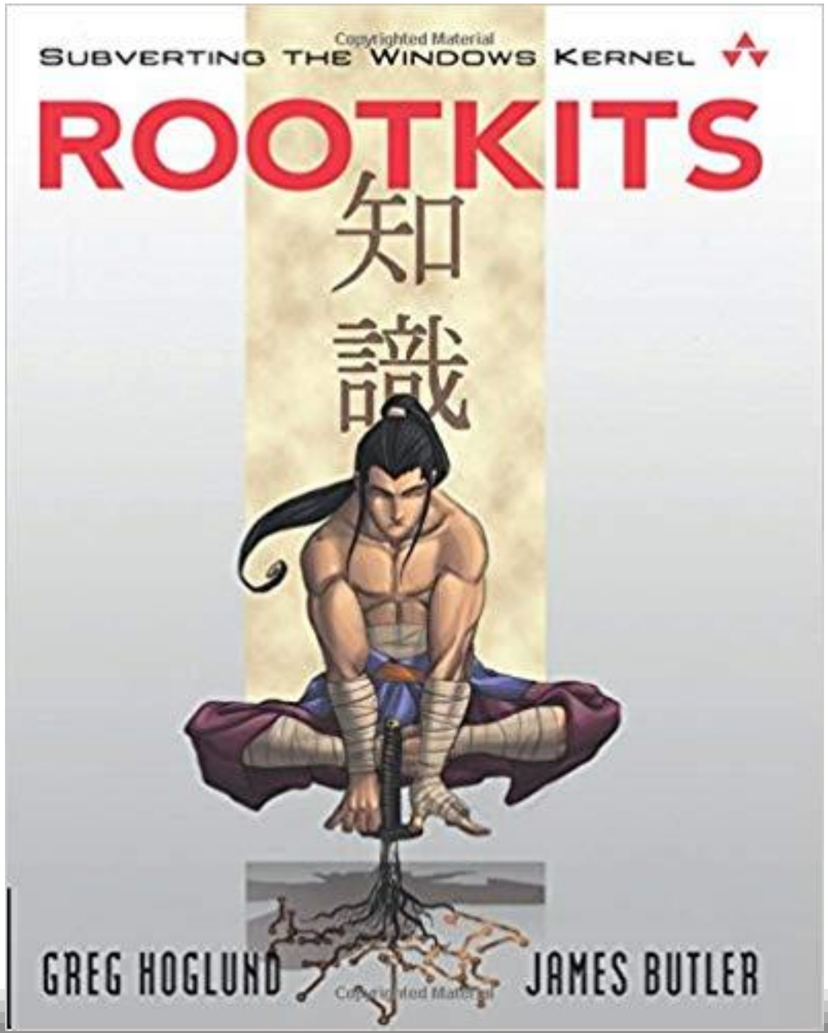
Matt Oh (jeongoh@Microsoft.com)

Microsoft

Whoami?

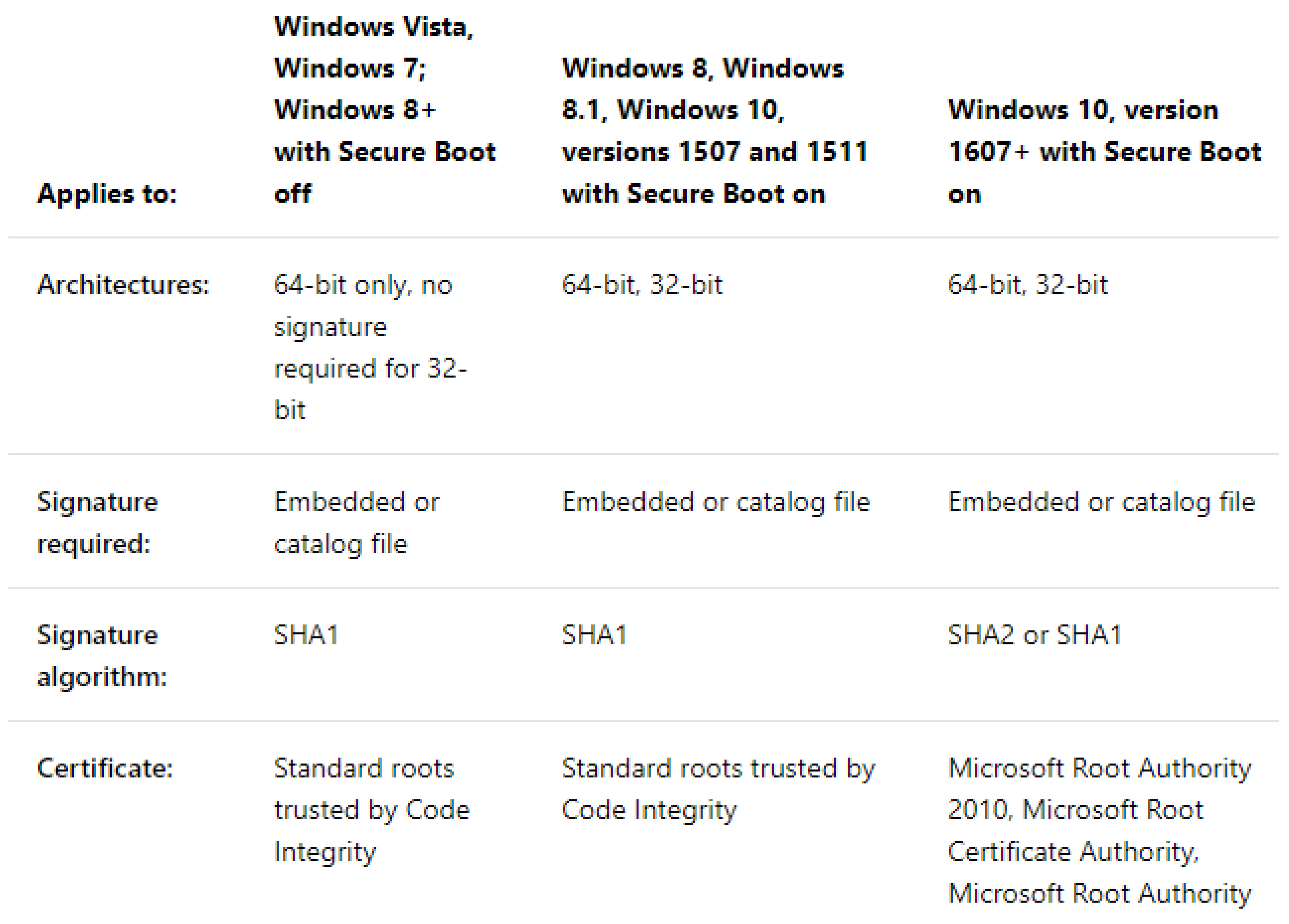
- Microsoft WDATP research team
  - EDR, Blueteam
  - Special interests in new exploit and malware technique
- Reverse engineer
  - Tearing down exploits, malware
  - Use the knowledge for better defense tactics/strategy
- 1-day researcher
  - DarunGrim – opensource binary diffing tool

Rootkits: Subverting the Windows Kernel



- Rootkits: Subverting the Windows Kernel was published in 2005
- This is **the** reference for Windows rootkits
- Many techniques were used by malware in the wild (DKOM, SSDT hooks)

Windows driver signing requirements



Applies to:	Windows Vista, Windows 7; Windows 8+ with Secure Boot off	Windows 8, Windows 8.1, Windows 10, versions 1507 and 1511 with Secure Boot on	Windows 10, version 1607+ with Secure Boot on
Architectures:	64-bit only, no signature required for 32-bit	64-bit, 32-bit	64-bit, 32-bit
Signature required:	Embedded or catalog file	Embedded or catalog file	Embedded or catalog file
Signature algorithm:	SHA1	SHA1	SHA2 or SHA1
Certificate:	Standard roots trusted by Code Integrity	Standard roots trusted by Code Integrity	Microsoft Root Authority 2010, Microsoft Root Certificate Authority, Microsoft Root Authority

https://docs.microsoft.com/en-us/windows-hardware/drivers/install/kernel-mode-code-signing-policy--windows-vista-and-later-

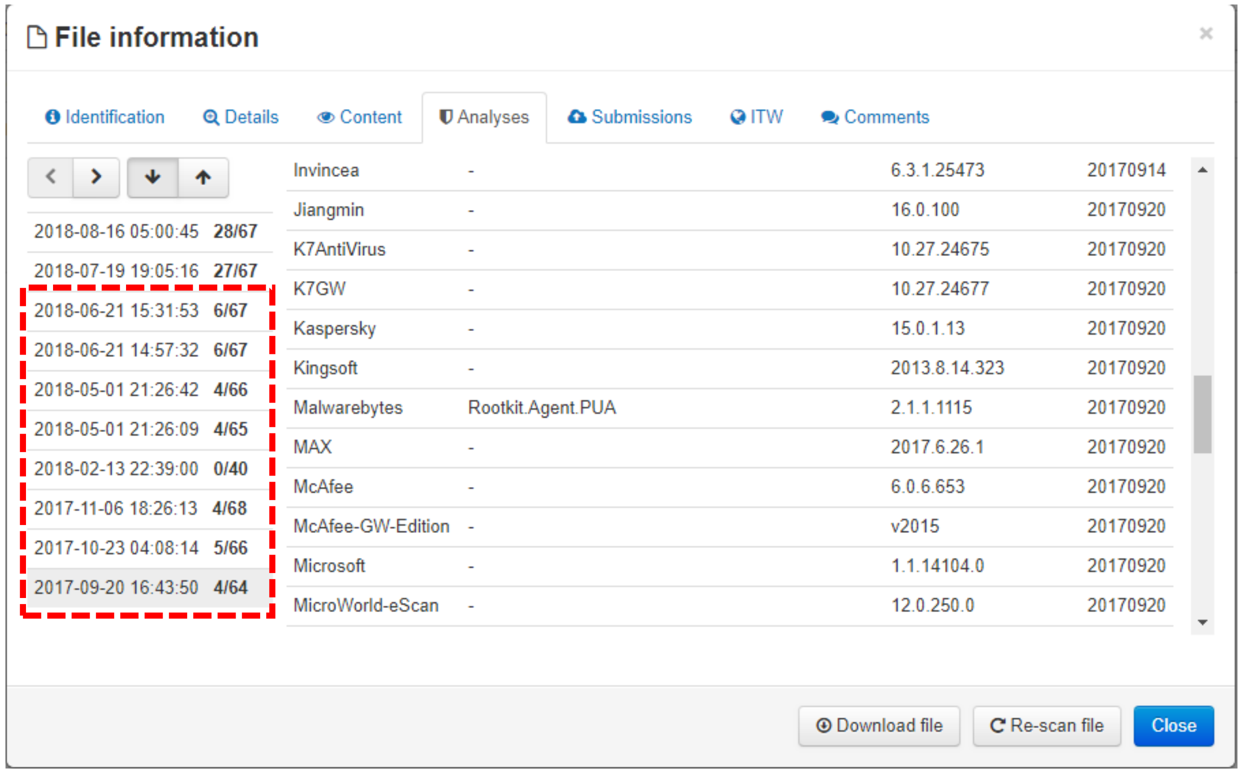
- The Windows rootkit era ended with the release of Windows Vista, mainly due to Windows signing requirements and Kernel Patch Protection (aka KPP, PatchGuard)
- Now malware authors need to overcome signing requirement
- Only very advanced actors used rootkits so far (Equation, Duqu2, etc)

Zacinlo ad fraud operation

- Bitdefender reported *Zacinlo Ad Fraud operation* in June of 2018
  - Microsoft detection – *Trojan:Win64/Detrahere*
  - Seemed very usual until it mentioned “digitally signed rootkit”
  - The report focused on the *ad fraud* aspects of the malware
- This presentation will focus on the rootkit aspects of the *Zacinlo* malware

Detrahere: low detections

- It is believed that the threat has been running since early 2012
- According to VirusTotal, Malwarebytes identified the rootkit component of this threat as early as September 2017
- Low detection rates (6/67) until June 2018



File information

2018-08-16 05:00:45	28/67
2018-07-19 19:05:16	27/67
2018-06-21 15:31:53	6/67
2018-06-21 14:57:32	6/67
2018-05-01 21:26:42	4/66
2018-05-01 21:26:09	4/65
2018-02-13 22:39:00	0/40
2017-11-06 18:26:13	4/68
2017-10-23 04:08:14	5/66
2017-09-20 16:43:50	4/64

Invincea	-	6.3.1.25473	20170914
Jiangmin	-	16.0.100	20170920
K7AntiVirus	-	10.27.24675	20170920
K7GW	-	10.27.24677	20170920
Kaspersky	-	15.0.1.13	20170920
Kingsoft	-	2013.8.14.323	20170920
Malwarebytes	Rootkit.Agent.PUA	2.1.1.1115	20170920
MAX	-	2017.6.26.1	20170920
McAfee	-	6.0.6.653	20170920
McAfee-GW-Edition	-	v2015	20170920
Microsoft	-	1.1.14104.0	20170920
MicroWorld-eScan	-	12.0.250.0	20170920

Detrahere: Stealthiness+Persistency

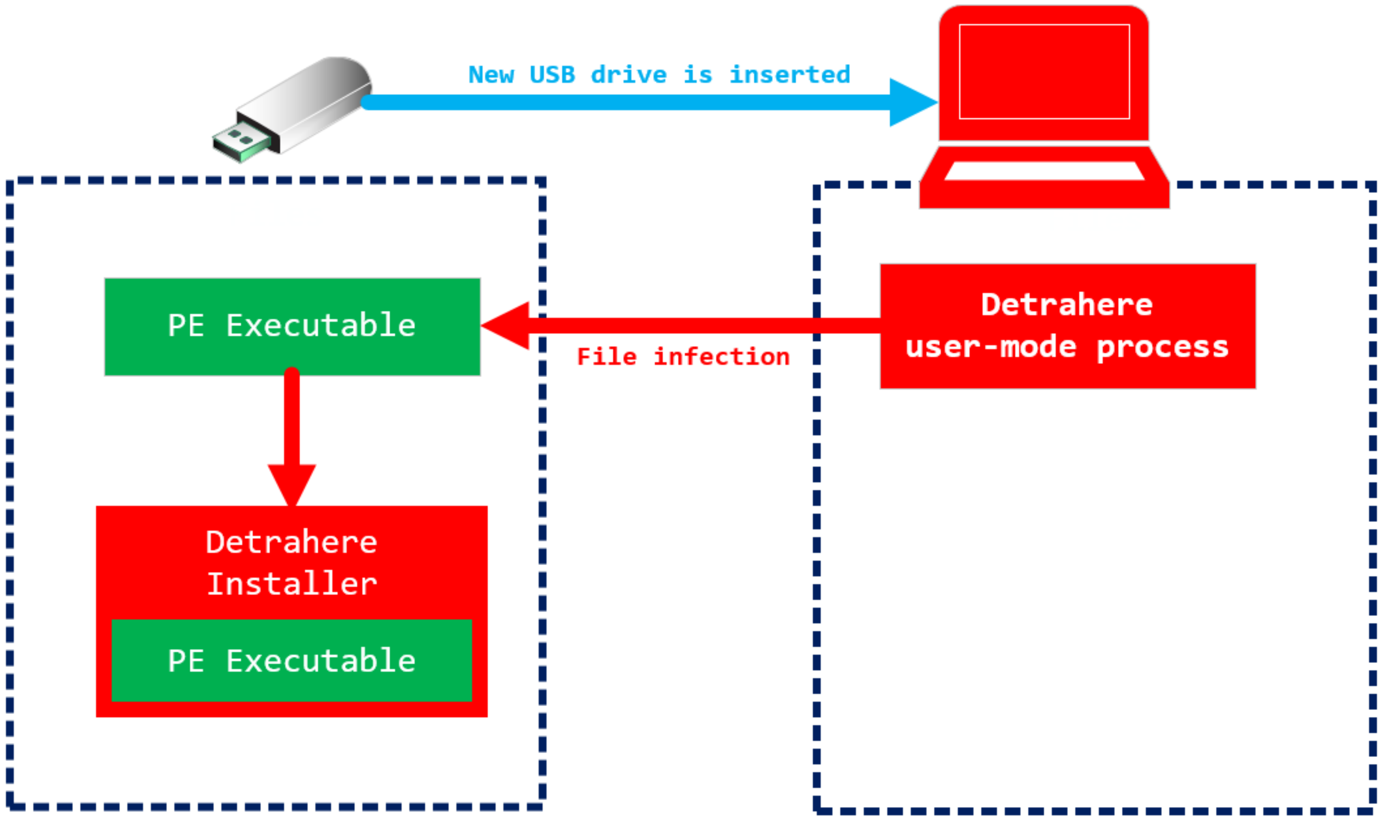
- The threat was under the radar for a long time
  - It infects other executable to propagate
    - The infected file will run the original executable after infecting victim machine
  - It installs a kernel driver that loads additional payload drivers from the hidden file system
    - The rootkit component blocks visibility into the related malware files using hidden file system
  - It registered it as a shutdown handler and also put itself in the early phase of driver loading order
    - Remediation can be challenging because it installs a shutdown handler to reinstall itself for persistency

Components

Name	Functionality	Descriptions
DriverProtect	Shutdown handler registration	Register a shutdown handler to regain persistence when the service is removed
	Hidden file system	Hide file contents of the malware files
	Anti-analysis/debugging	Block security products and analysts tools process launch and check for attached kernel debugger
User-mode process	USB file infection	When a USB drive is connected, all PE files on it will be infected
	Network traffic injection	Modify network traffic and inject Ad Fraud
	C&C	Connect to C&C servers
Netfilter2	Network traffic injection (driver)	Provides driver-level support for network traffic injection which will be used by user-mode process
udiskMgr	Anti-remediation	Blocks security products and analysts tools process launch Blocks some files creation (ex. FIXLISTS.TXT) used by a system recovery tool

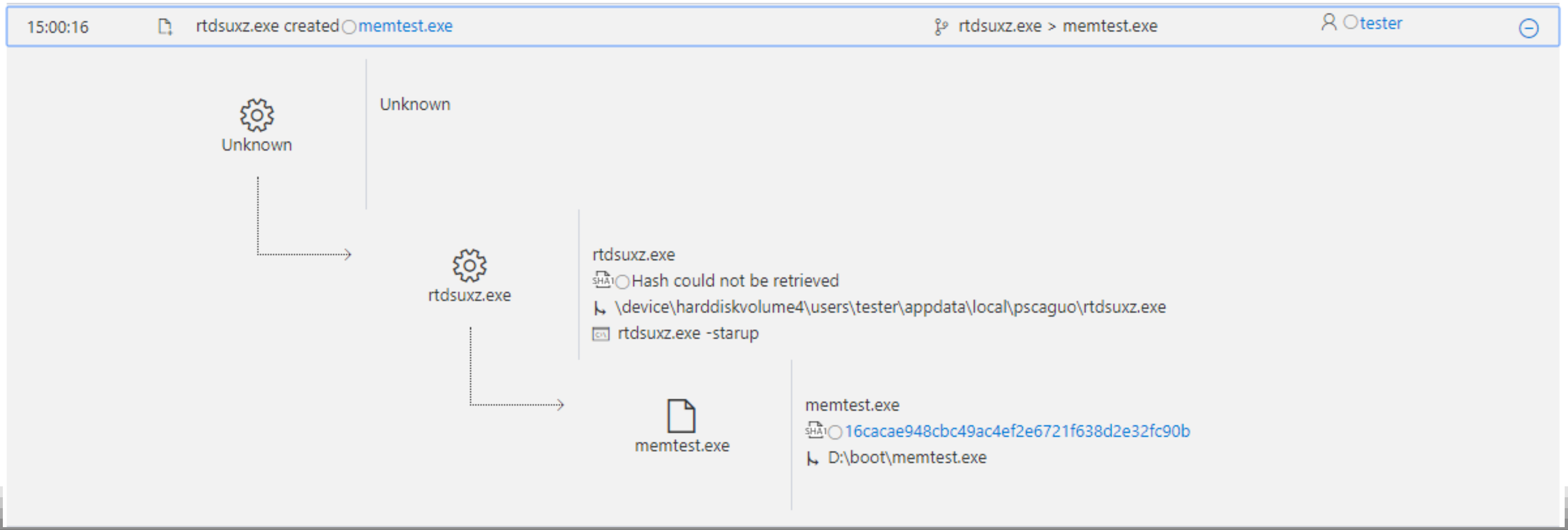
Infection/propagation

USB file infection: threat delivery mechanism



USB file infection: threat delivery mechanism

- Bitdefender report focused on one case where malware is delivered through fake VPN client.
- In reality, the threat can spread through USB infections. When a user inserts a USB drive into the infected machine, the rootkit component will infect the PE files on the USB drive
  - Probably the VPN client found by Bitdefender might be infected in the first place.



15:00:16 rtdsuxz.exe created memtest.exe rtdsuxz.exe > memtest.exe tester

Unknown

Unknown

rtdsuxz.exe
Hash could not be retrieved
\device\harddiskvolume4\users\tester\appdata\local\pscaguo\rtdsuxz.exe
rtdsuxz.exe -starup

memtest.exe
16cacae948cbc49ac4ef2e6721f638d2e32fc90b
D:\boot\memtest.exe

File infection

```
Verified:        Signed
Signing date:    2:45 AM 7/16/2016
Publisher:       Microsoft Windows
Company:         Microsoft Corporation
Description:     Notepad
Product:         Microsoft« Windows«
Operating System
Prod version:    10.0.14393.0
File version:    10.0.14393.0
(rs1_release.160715-1616)
MachineType:     64-bit
```

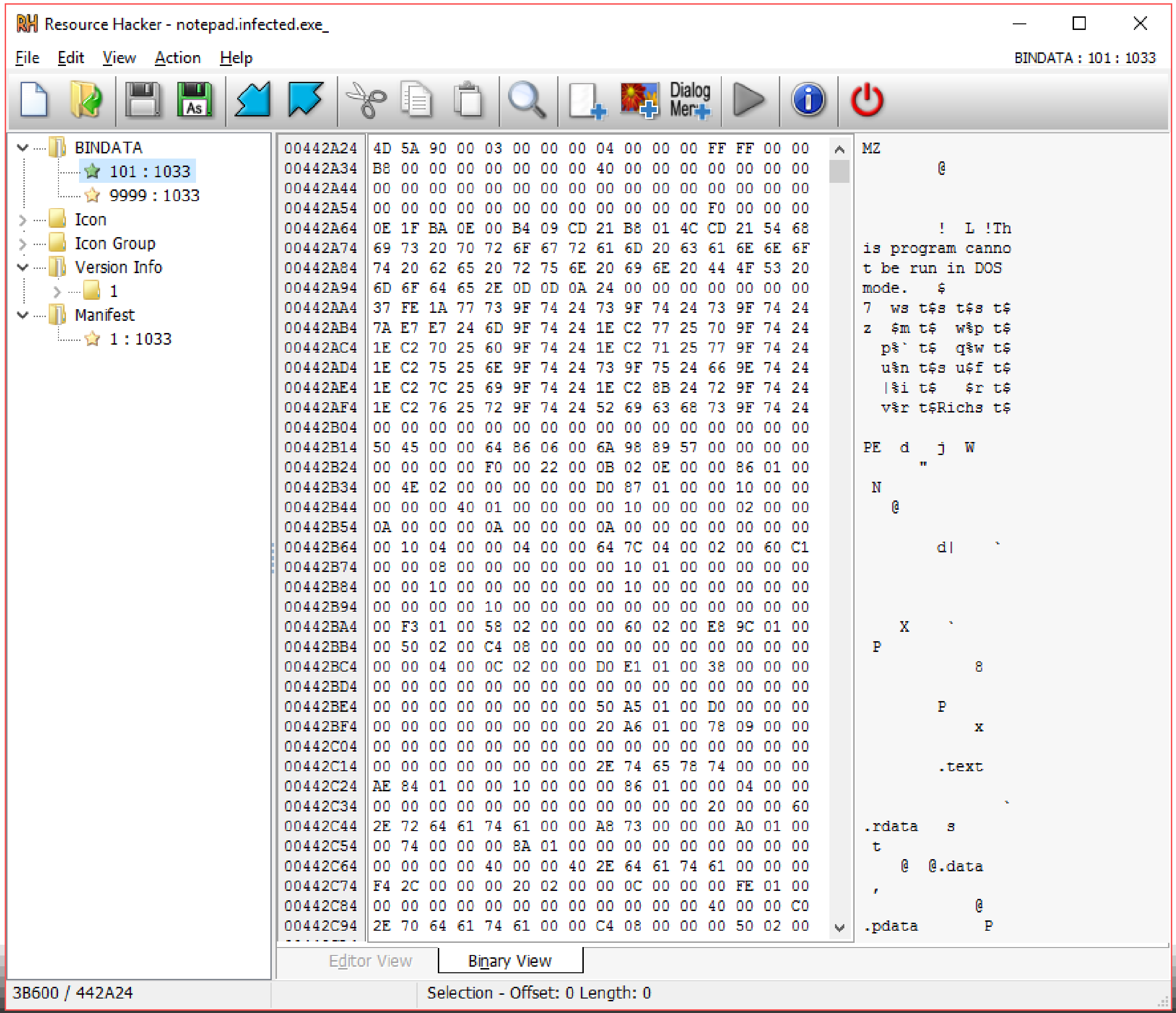
Infection

```
Verified:        Unsigned
Link date:       5:24 AM 11/21/2017
Publisher:       n/a
Company:         Microsoft Corporation
Description:     Notepad
Product:         Microsoft« Windows« Operating
System
Prod version:    10.0.14393.0
File version:    10.0.14393.0
(rs1_release.160715-1616)
MachineType:     32-bit
```

The file becomes unsigned and the machine type becomes 32-bit because the malware only has a 32-bit infector.

The original 64-bit file will run by this infector later after being extracted from resource section of the infected file.

Original PE file is inserted as a resource

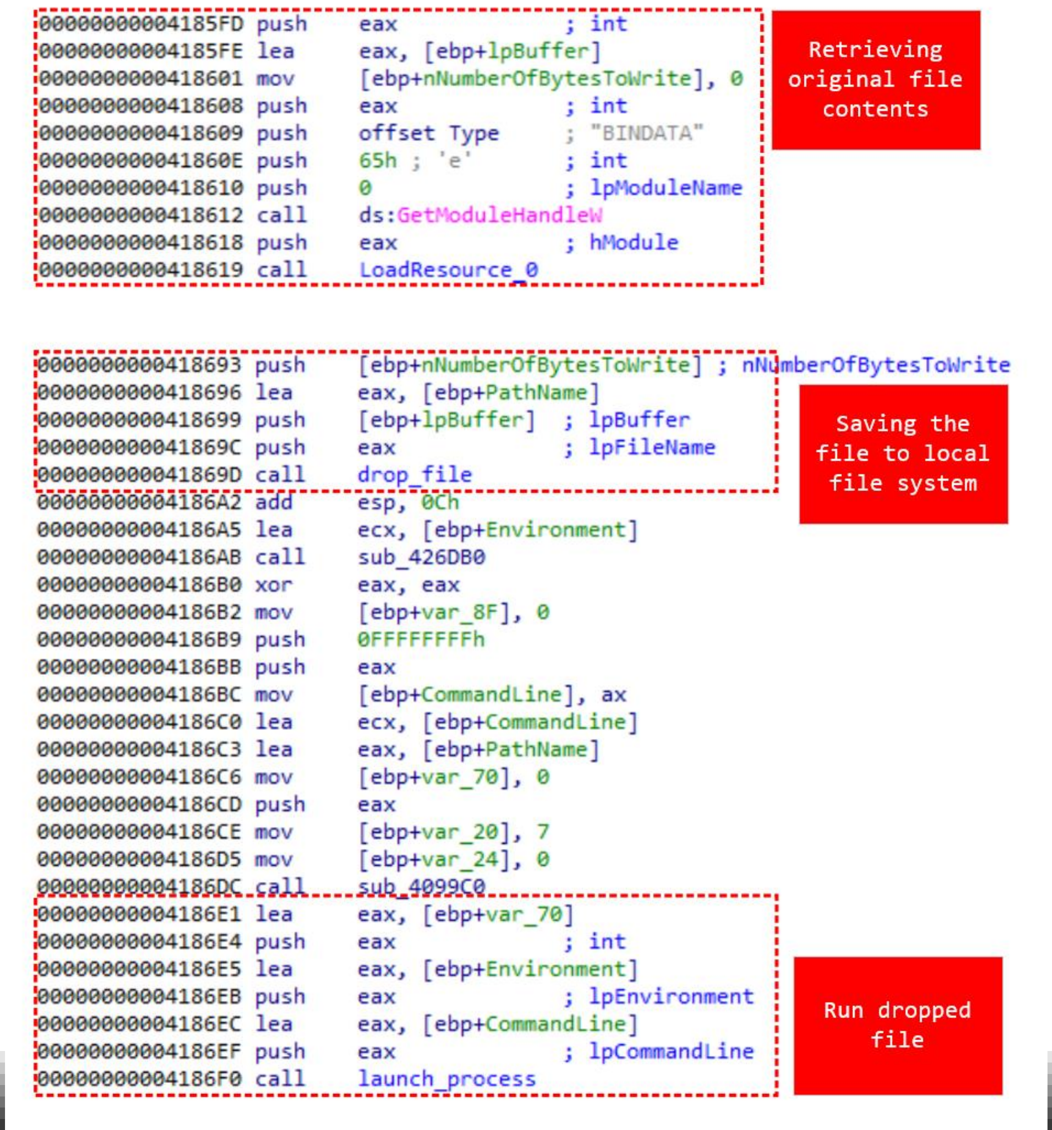


```
00442A24  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00   MZ
00442A34  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00           @
00442A44  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00442A54  00 00 00 00 00 00 00 00 00 00 00 00 F0 00 00 00
00442A64  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68           !  L !Th
00442A74  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F   is program canno
00442A84  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20   t be run in DOS
00442A94  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00   mode.    $
00442AA4  37 FE 1A 77 73 9F 74 24 73 9F 74 24 73 9F 74 24   7  ws t$s t$s t$
00442AB4  7A E7 E7 24 6D 9F 74 24 1E C2 77 25 70 9F 74 24   z  $m t$  w%p t$
00442AC4  1E C2 70 25 60 9F 74 24 1E C2 71 25 77 9F 74 24     p%` t$  q%w t$
00442AD4  1E C2 75 25 6E 9F 74 24 73 9F 75 24 66 9E 74 24     u%n t$s u$f t$
00442AE4  1E C2 7C 25 69 9F 74 24 1E C2 8B 24 72 9F 74 24     |%i t$   $r t$
00442AF4  1E C2 76 25 72 9F 74 24 52 69 63 68 73 9F 74 24     v%r t$Richs t$
00442B04  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00442B14  50 45 00 00 64 86 06 00 6A 98 89 57 00 00 00 00   PE  d   j  W
00442B24  00 00 00 00 F0 00 22 00 0B 02 0E 00 00 86 01 00         "
00442B34  00 4E 02 00 00 00 00 00 D0 87 01 00 00 10 00 00    N
00442B44  00 00 00 40 01 00 00 00 00 10 00 00 00 02 00 00      @
00442B54  0A 00 00 00 0A 00 00 00 0A 00 00 00 00 00 00 00
00442B64  00 10 04 00 00 04 00 00 64 7C 04 00 02 00 60 C1           d|    `
00442B74  00 00 08 00 00 00 00 00 00 10 01 00 00 00 00 00
00442B84  00 00 10 00 00 00 00 00 00 10 00 00 00 00 00 00
00442B94  00 00 00 00 10 00 00 00 00 00 00 00 00 00 00 00
00442BA4  00 F3 01 00 58 02 00 00 00 60 02 00 E8 9C 01 00       X    `
00442BB4  00 50 02 00 C4 08 00 00 00 00 00 00 00 00 00 00    P
00442BC4  00 00 04 00 0C 02 00 00 D0 E1 01 00 38 00 00 00               8
00442BD4  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00442BE4  00 00 00 00 00 00 00 00 50 A5 01 00 D0 00 00 00           P
00442BF4  00 00 00 00 00 00 00 00 20 A6 01 00 78 09 00 00               x
00442C04  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00442C14  00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00           .text
00442C24  AE 84 01 00 00 10 00 00 00 86 01 00 00 04 00 00
00442C34  00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 60                  `
00442C44  2E 72 64 61 74 61 00 00 A8 73 00 00 00 A0 01 00   .rdata   s
00442C54  00 74 00 00 00 8A 01 00 00 00 00 00 00 00 00 00    t
00442C64  00 00 00 00 40 00 00 40 2E 64 61 74 61 00 00 00       @  @.data
00442C74  F4 2C 00 00 00 20 02 00 00 0C 00 00 00 FE 01 00    ,
00442C84  00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 C0               @
00442C94  2E 70 64 61 74 61 00 00 C4 08 00 00 00 50 02 00   .pdata       P
```

3B600 / 442A24

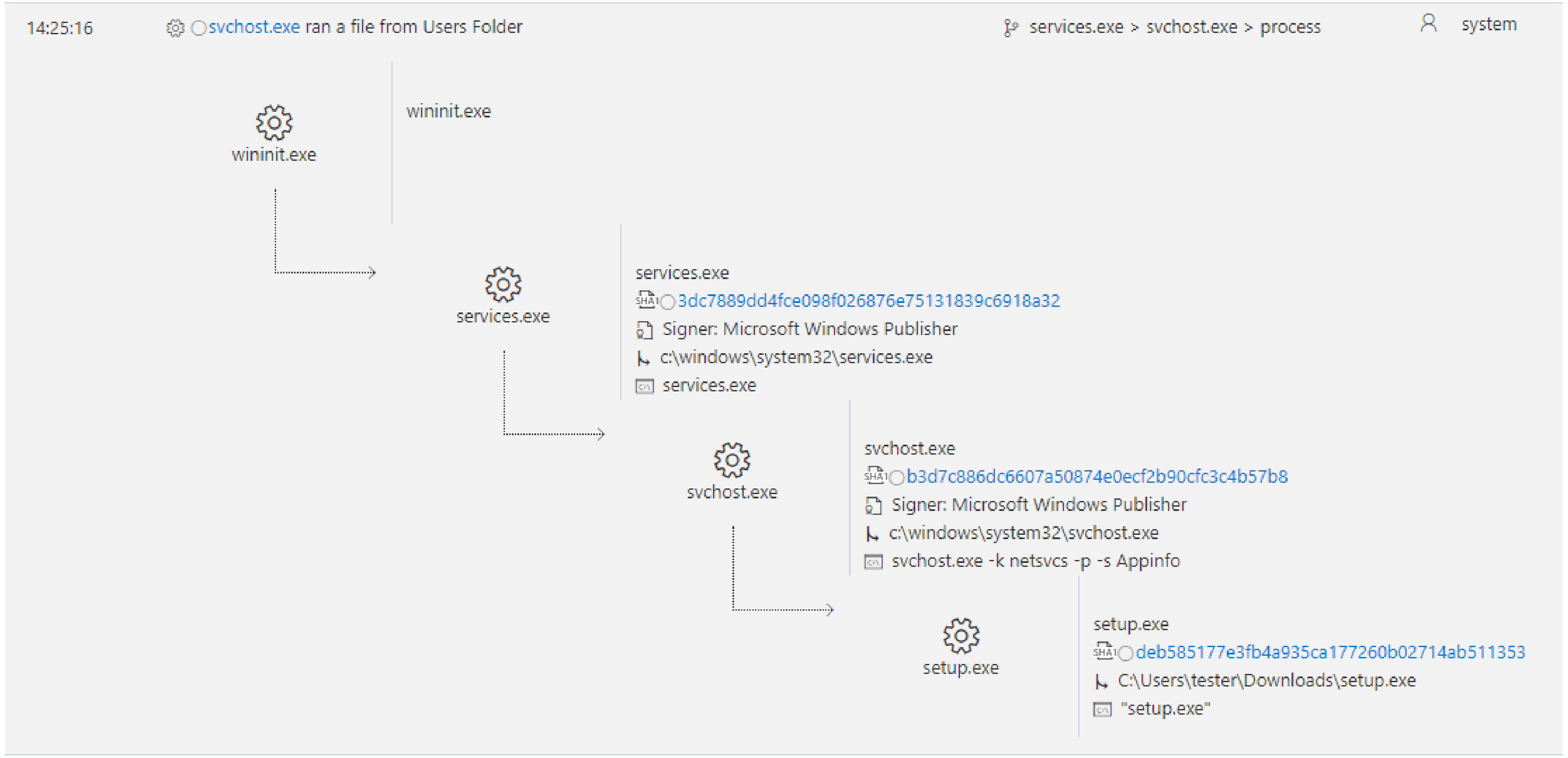
Selection - Offset: 0 Length: 0

Running original PE payload


Retrieving original file contents
Saving the file to local file system
Run dropped file

```
00000000004185FD push    eax                   ; int
00000000004185FE lea     eax, [ebp+lpBuffer]
0000000000418601 mov     [ebp+nNumberOfBytesToWrite], 0
0000000000418608 push    eax                   ; int
0000000000418609 push    offset Type           ; "BINDATA"
000000000041860E push    65h ; 'e'             ; int
0000000000418610 push    0                     ; lpModuleName
0000000000418612 call    ds:GetModuleHandleW
0000000000418618 push    eax                   ; hModule
0000000000418619 call    LoadResource_0

0000000000418693 push    [ebp+nNumberOfBytesToWrite] ; nNumberOfBytesToWrite
0000000000418696 lea     eax, [ebp+PathName]
0000000000418699 push    [ebp+lpBuffer]  ; lpBuffer
000000000041869C push    eax             ; lpFileName
000000000041869D call    drop_file
00000000004186A2 add     esp, 0Ch
00000000004186A5 lea     ecx, [ebp+Environment]
00000000004186AB call    sub_426DB0
00000000004186B0 xor     eax, eax
00000000004186B2 mov     [ebp+var_8F], 0
00000000004186B9 push    0FFFFFFFFh
00000000004186BB push    eax
00000000004186BC mov     [ebp+CommandLine], ax
00000000004186C0 lea     ecx, [ebp+CommandLine]
00000000004186C3 lea     eax, [ebp+PathName]
00000000004186C6 mov     [ebp+var_70], 0
00000000004186CD push    eax
00000000004186CE mov     [ebp+var_20], 7
00000000004186D5 mov     [ebp+var_24], 0
00000000004186DC call    sub_4099C0
00000000004186E1 lea     eax, [ebp+var_70]
00000000004186E4 push    eax                   ; int
00000000004186E5 lea     eax, [ebp+Environment]
00000000004186EB push    eax                   ; lpEnvironment
00000000004186EC lea     eax, [ebp+CommandLine]
00000000004186EF push    eax                   ; lpCommandLine
00000000004186F0 call    launch_process
```

Running the infected executable



14:25:16 svchost.exe ran a file from Users Folder

services.exe > svchost.exe > process

system

wininit.exe

wininit.exe

services.exe

services.exe
3dc7889dd4fce098f026876e75131839c6918a32
Signer: Microsoft Windows Publisher
c:\windows\system32\services.exe
services.exe

svchost.exe

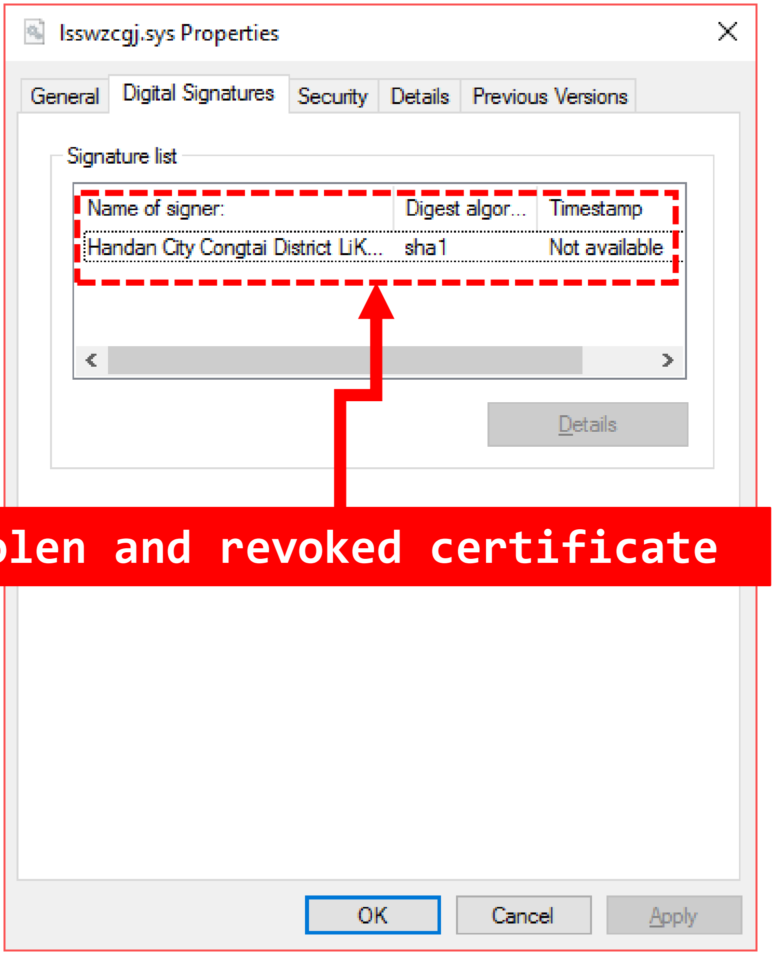
svchost.exe
b3d7c886dc6607a50874e0ecf2b90cfc3c4b57b8
Signer: Microsoft Windows Publisher
c:\windows\system32\svchost.exe
svchost.exe -k netsvcs -p -s Appinfo

setup.exe

setup.exe
deb585177e3fb4a935ca177260b02714ab511353
C:\Users\tester\Downloads\setup.exe
"setup.exe"

Loading kernel driver

Stolen and revoked certificate



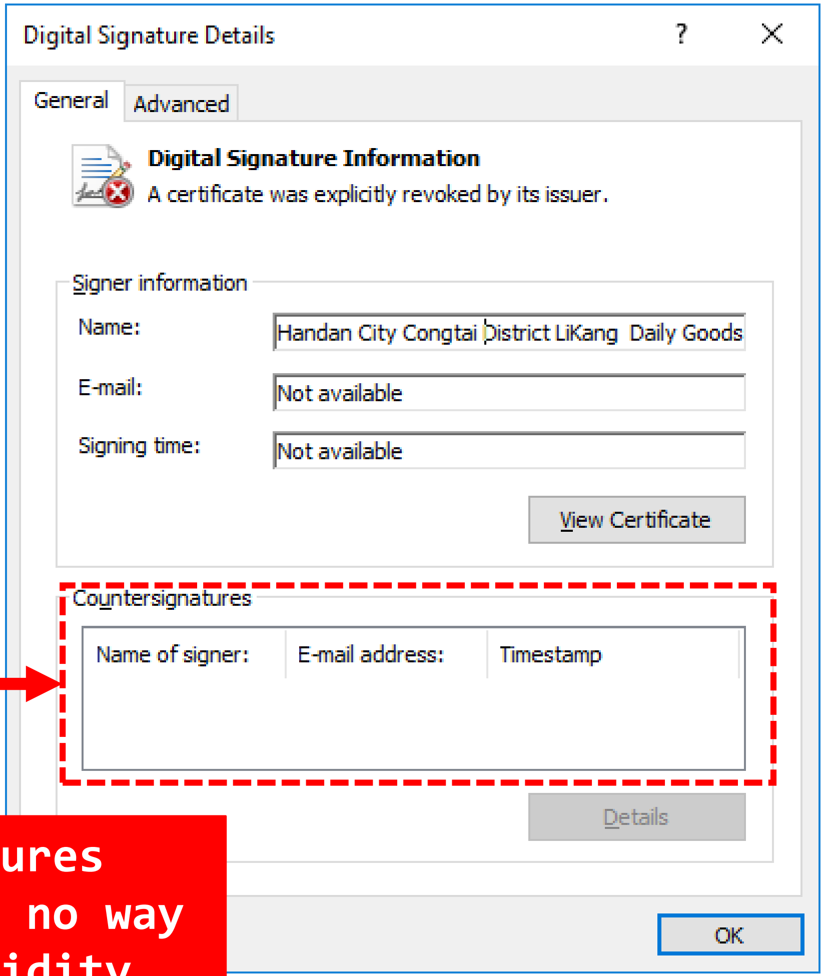
lsswzcgj.sys Properties

General | Digital Signatures | Security | Details | Previous Versions

Signature list

Name of signer:	Digest algor...	Timestamp
Handan City Congtai District LiK...	sha1	Not available

Stolen and revoked certificate



Digital Signature Details

General | Advanced

Digital Signature Information
A certificate was explicitly revoked by its issuer.

Signer information

Name: Handan City Congtai District LiKang Daily Goods

E-mail: Not available

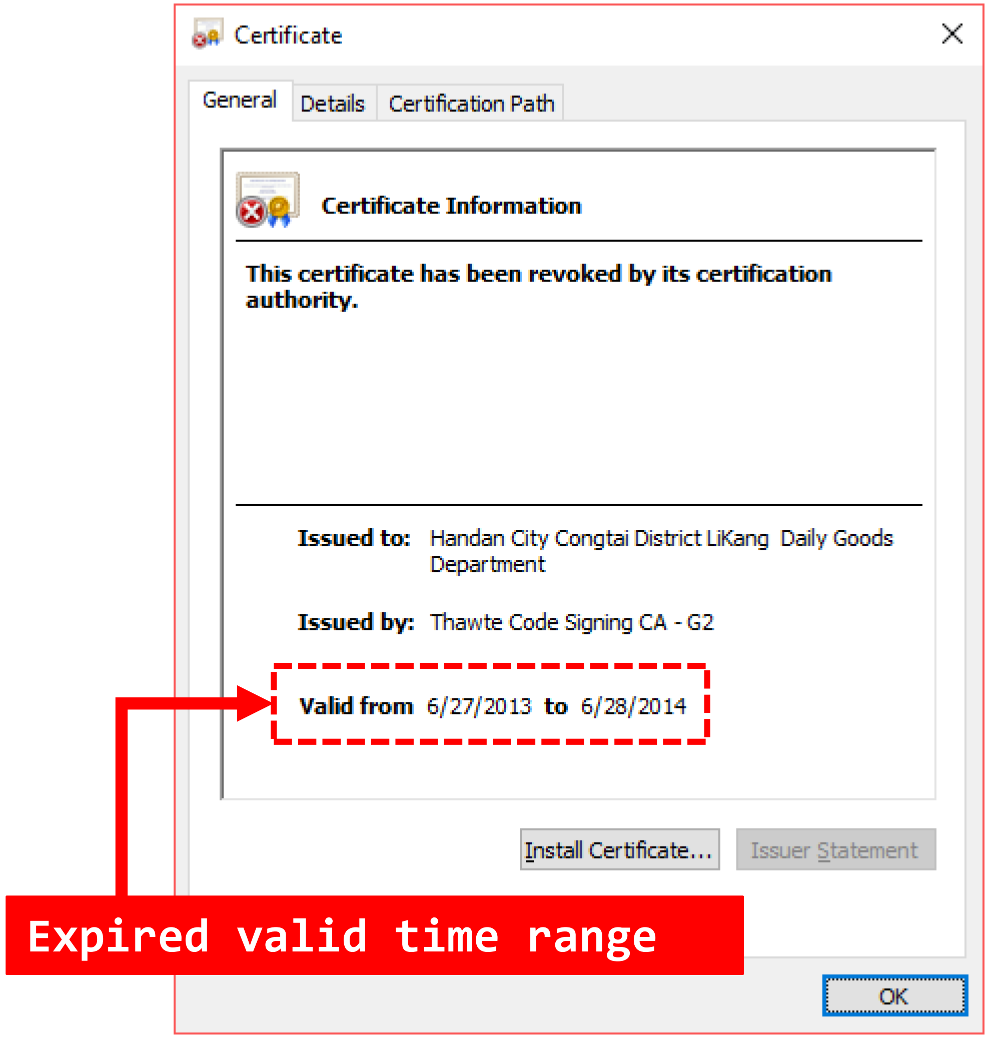
Signing time: Not available

Countersignatures

Name of signer:	E-mail address:	Timestamp

Countersignatures are missing – no way to verify validity of the certificate

And expired



Certificate

Certificate Information

This certificate has been revoked by its certification authority.

Issued to: Handan City Congtai District LiKang Daily Goods Department

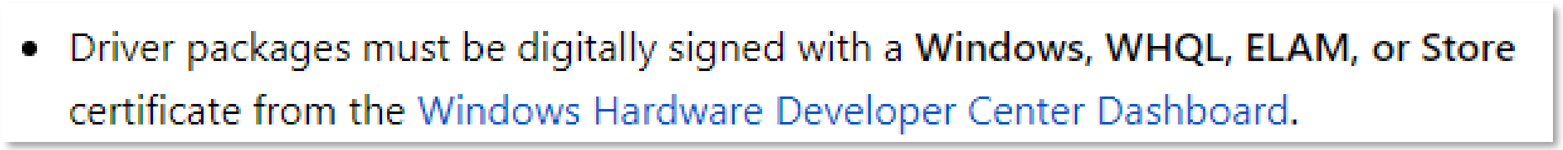
Issued by: Thawte Code Signing CA - G2

Valid from 6/27/2013 **to** 6/28/2014

- The certificate used for this driver was issued to "*Handan City Congtai District LiKang Daily Goods Department*" and was revoked
- The malware performs system time change to sign this driver on-the-fly to generate expired certificate intentionally
- For compatibility reasons, Windows will accept this certificate thinking it was previously generated for legacy driver

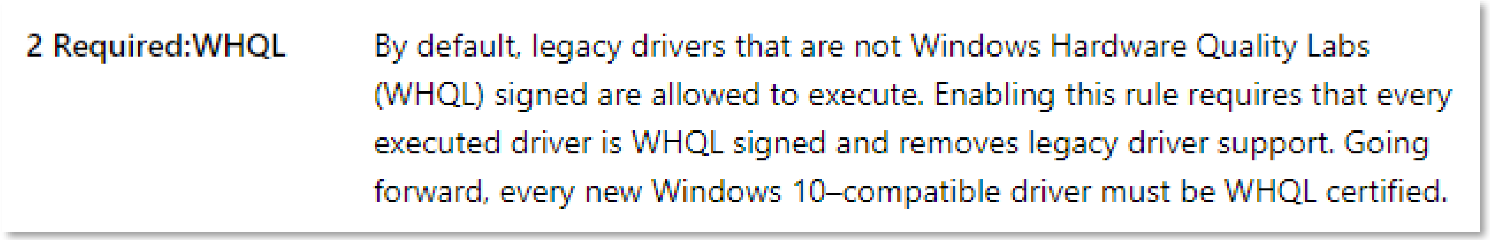
Mitigations

- Windows 10 S mode will prevent loading of these drivers
  - More strict driver requirements:



- Driver packages must be digitally signed with a **Windows, WHQL, ELAM, or Store** certificate from the Windows Hardware Developer Center Dashboard.

- Windows Defender Application Control policy (Device Guard)



2 Required:WHQL	By default, legacy drivers that are not Windows Hardware Quality Labs (WHQL) signed are allowed to execute. Enabling this rule requires that every executed driver is WHQL signed and removes legacy driver support. Going forward, every new Windows 10–compatible driver must be WHQL certified.

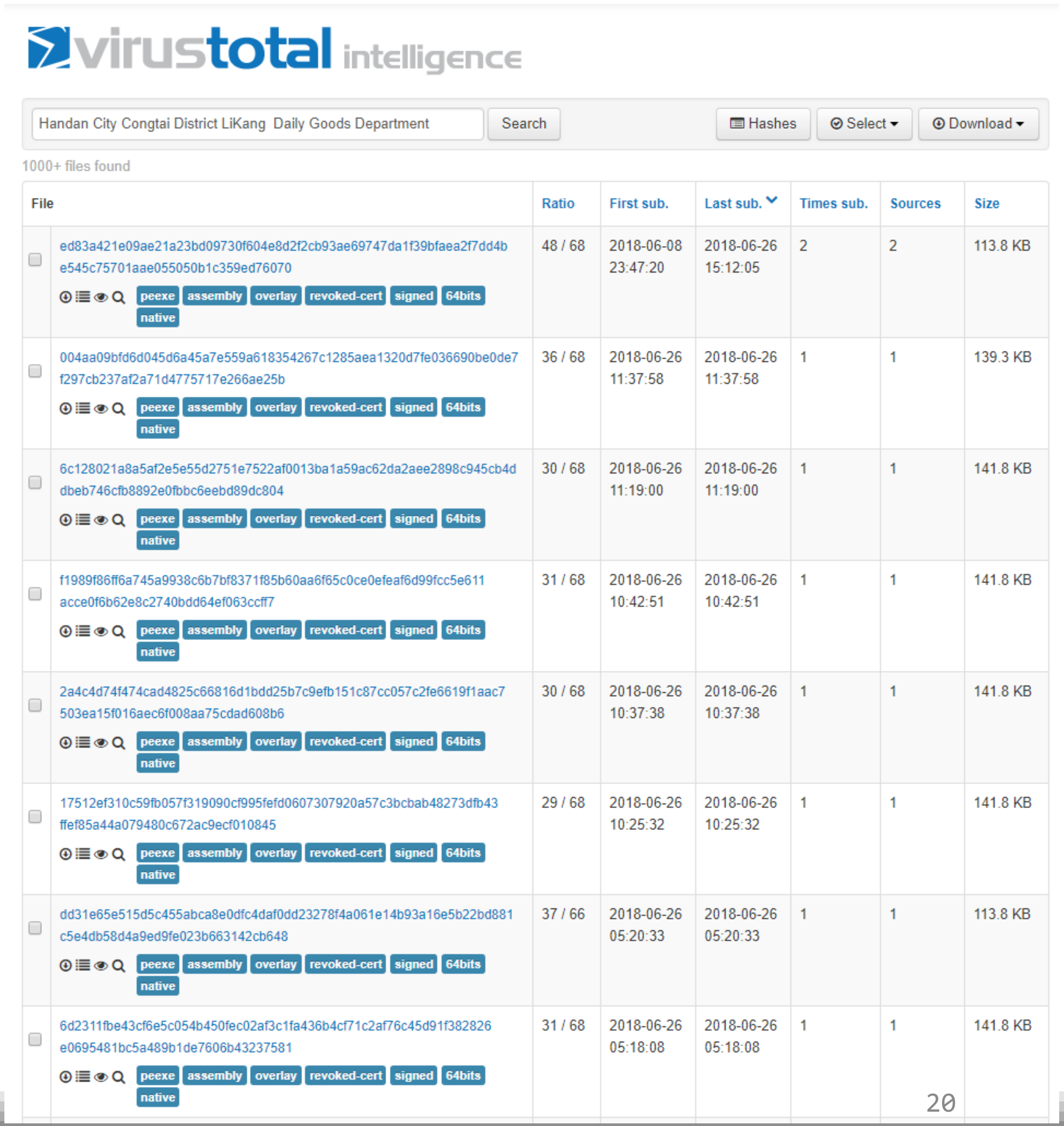
- SecureBoot + HVCI (Memory Integrity) + VBS
  - More of anti-rootkit, exploit approach

Detections: WDATP

- RS5 detection: certificate telemetry + machine learning
  - The revoked and expired certificate from a vendor never signed Windows kernel driver
  - Using machine learning to mass analyze the certificate information
  - When it fits into the profile, detection will be made

Variant drivers

- Searching VTI (Virus Total Intelligence) will return thousands of files with revoked certificate from "*Handan City Congtai District LiKang Daily Goods Department*"
- These are all variants files generated by the infector

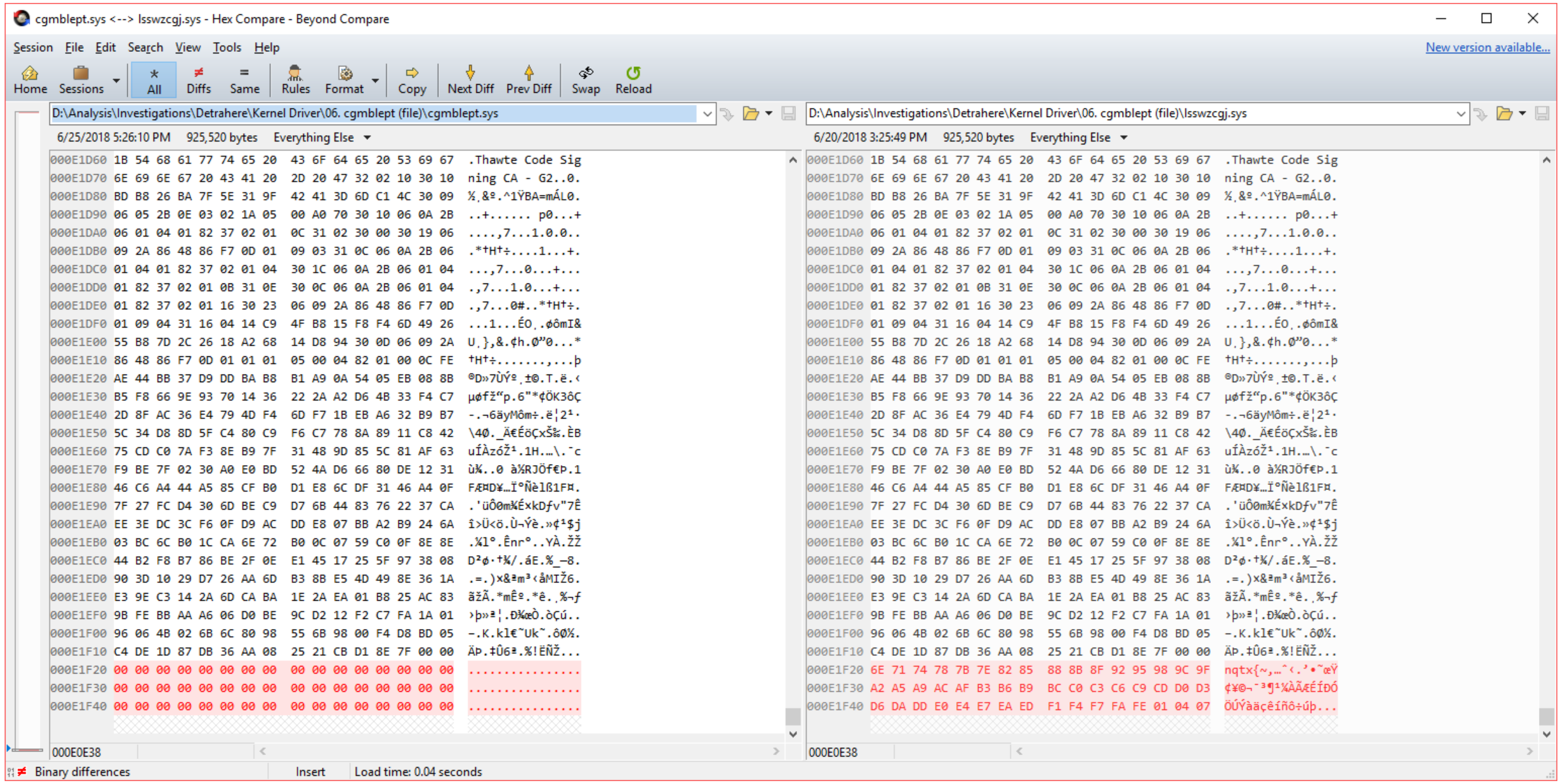


1000+ files found

File	Ratio	First sub.	Last sub.	Times sub.	Sources	Size
ed83a421e09ae21a23bd09730f604e8d2f2cb93ae69747da1f39bfaea2f7dd4b e545c75701aae055050b1c359ed76070 peexe assembly overlay revoked-cert signed 64bits native	48 / 68	2018-06-08 23:47:20	2018-06-26 15:12:05	2	2	113.8 KB
004aa09bfd6d045d6a45a7e559a618354267c1285aea1320d7fe036690be0de7 f297cb237af2a71d4775717e266ae25b peexe assembly overlay revoked-cert signed 64bits native	36 / 68	2018-06-26 11:37:58	2018-06-26 11:37:58	1	1	139.3 KB
6c128021a8a5af2e5e55d2751e7522af0013ba1a59ac62da2aee2898c945cb4d dbeb746cfb8892e0fbbc6eebd89dc804 peexe assembly overlay revoked-cert signed 64bits native	30 / 68	2018-06-26 11:19:00	2018-06-26 11:19:00	1	1	141.8 KB
f1989f86ff6a745a9938c6b7bf8371f85b60aa6f65c0ce0efeaf6d99fcc5e611 acce0f6b62e8c2740bdd64ef063ccff7 peexe assembly overlay revoked-cert signed 64bits native	31 / 68	2018-06-26 10:42:51	2018-06-26 10:42:51	1	1	141.8 KB
2a4c4d74f474cad4825c66816d1bdd25b7c9efb151c87cc057c2fe6619f1aac7 503ea15f016aec6f008aa75cdad608b6 peexe assembly overlay revoked-cert signed 64bits native	30 / 68	2018-06-26 10:37:38	2018-06-26 10:37:38	1	1	141.8 KB
17512ef310c59fb057f319090cf995fefd0607307920a57c3bcbab48273dfb43 ffef85a44a079480c672ac9ecf010845 peexe assembly overlay revoked-cert signed 64bits native	29 / 68	2018-06-26 10:25:32	2018-06-26 10:25:32	1	1	141.8 KB
dd31e65e515d5c455abca8e0dfc4daf0dd23278f4a061e14b93a16e5b22bd881 c5e4db58d4a9ed9fe023b663142cb648 peexe assembly overlay revoked-cert signed 64bits native	37 / 66	2018-06-26 05:20:33	2018-06-26 05:20:33	1	1	113.8 KB
6d2311fbe43cf6e5c054b450fec02af3c1fa436b4cf71c2af76c45d91f382826 e0695481bc5a489b1de7606b43237581 peexe assembly overlay revoked-cert signed 64bits native	31 / 68	2018-06-26 05:18:08	2018-06-26 05:18:08	1	1	141.8 KB

Variant drivers

Variants are basically same except some padded bytes


cgmblept.sys <--> lsswzcgj.sys - Hex Compare - Beyond Compare
Session File Edit Search View Tools Help
New version available...
Home Sessions All Diffs Same Rules Format Copy Next Diff Prev Diff Swap Reload

D:\Analysis\Investigations\Detrahere\Kernel Driver\06. cgmblept (file)\cgmblept.sys

6/25/2018 5:26:10 PM 925,520 bytes Everything Else

```
000E1D60 1B 54 68 61 77 74 65 20  43 6F 64 65 20 53 69 67  .Thawte Code Sig
000E1D70 6E 69 6E 67 20 43 41 20  2D 20 47 32 02 10 30 10  ning CA - G2..0.
000E1D80 BD B8 26 BA 7F 5E 31 9F  42 41 3D 6D C1 4C 30 09  ½¸&º.^1ŸBA=mÁL0.
000E1D90 06 05 2B 0E 03 02 1A 05  00 A0 70 30 10 06 0A 2B  ..+...... p0...+
000E1DA0 06 01 04 01 82 37 02 01  0C 31 02 30 00 30 19 06  ....‚7...1.0.0..
000E1DB0 09 2A 86 48 86 F7 0D 01  09 03 31 0C 06 0A 2B 06  .*†H†÷....1...+.
000E1DC0 01 04 01 82 37 02 01 04  30 1C 06 0A 2B 06 01 04  ...‚7...0...+...
000E1DD0 01 82 37 02 01 0B 31 0E  30 0C 06 0A 2B 06 01 04  .‚7...1.0...+...
000E1DE0 01 82 37 02 01 16 30 23  06 09 2A 86 48 86 F7 0D  .‚7...0#..*†H†÷.
000E1DF0 01 09 04 31 16 04 14 C9  4F B8 15 F8 F4 6D 49 26  ...1...ÉO¸.øômI&
000E1E00 55 B8 7D 2C 26 18 A2 68  14 D8 94 30 0D 06 09 2A  U¸},&.¢h.Ø”0...*
000E1E10 86 48 86 F7 0D 01 01 01  05 00 04 82 01 00 0C FE  †H†÷.......‚...þ
000E1E20 AE 44 BB 37 D9 DD BA B8  B1 A9 0A 54 05 EB 08 8B  ®D»7ÙÝº¸±©.T.ë.‹
000E1E30 B5 F8 66 9E 93 70 14 36  22 2A A2 D6 4B 33 F4 C7  µøfž“p.6"*¢ÖK3ôÇ
000E1E40 2D 8F AC 36 E4 79 4D F4  6D F7 1B EB A6 32 B9 B7  -.¬6äyMôm÷.ë¦2¹·
000E1E50 5C 34 D8 8D 5F C4 80 C9  F6 C7 78 8A 89 11 C8 42  \4Ø._Ä€ÉöÇxŠ‰.ÈB
000E1E60 75 CD C0 7A F3 8E B9 7F  31 48 9D 85 5C 81 AF 63  uÍÀzóŽ¹.1H.…\.¯c
000E1E70 F9 BE 7F 02 30 A0 E0 BD  52 4A D6 66 80 DE 12 31  ù¾..0 à½RJÖf€Þ.1
000E1E80 46 C6 A4 44 A5 85 CF B0  D1 E8 6C DF 31 46 A4 0F  FÆ¤D¥…Ï°Ñèlß1F¤.
000E1E90 7F 27 FC D4 30 6D BE C9  D7 6B 44 83 76 22 37 CA  .'üÔ0m¾É×kDƒv"7Ê
000E1EA0 EE 3E DC 3C F6 0F D9 AC  DD E8 07 BB A2 B9 24 6A  î>Ü<ö.Ù¬Ýè.»¢¹$j
000E1EB0 03 BC 6C B0 1C CA 6E 72  B0 0C 07 59 C0 0F 8E 8E  .¼l°.Ênr°..YÀ.ŽŽ
000E1EC0 44 B2 F8 B7 86 BE 2F 0E  E1 45 17 25 5F 97 38 08  D²ø·†¾/.áE.%_—8.
000E1ED0 90 3D 10 29 D7 26 AA 6D  B3 8B E5 4D 49 8E 36 1A  .=.)×&ªm³‹åMIŽ6.
000E1EE0 E3 9E C3 14 2A 6D CA BA  1E 2A EA 01 B8 25 AC 83  ãžÃ.*mÊº.*ê.¸%¬ƒ
000E1EF0 9B FE BB AA A6 06 D0 BE  9C D2 12 F2 C7 FA 1A 01  ›þ»ª¦.Ð¾œÒ.òÇú..
000E1F00 96 06 4B 02 6B 6C 80 98  55 6B 98 00 F4 D8 BD 05  –.K.kl€˜Uk˜.ôØ½.
000E1F10 C4 DE 1D 87 DB 36 AA 08  25 21 CB D1 8E 7F 00 00  ÄÞ.‡Û6ª.%!ËÑŽ...
000E1F20 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000E1F30 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
000E1F40 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

000E0E38

D:\Analysis\Investigations\Detrahere\Kernel Driver\06. cgmblept (file)\lsswzcgj.sys

6/20/2018 3:25:49 PM 925,520 bytes Everything Else

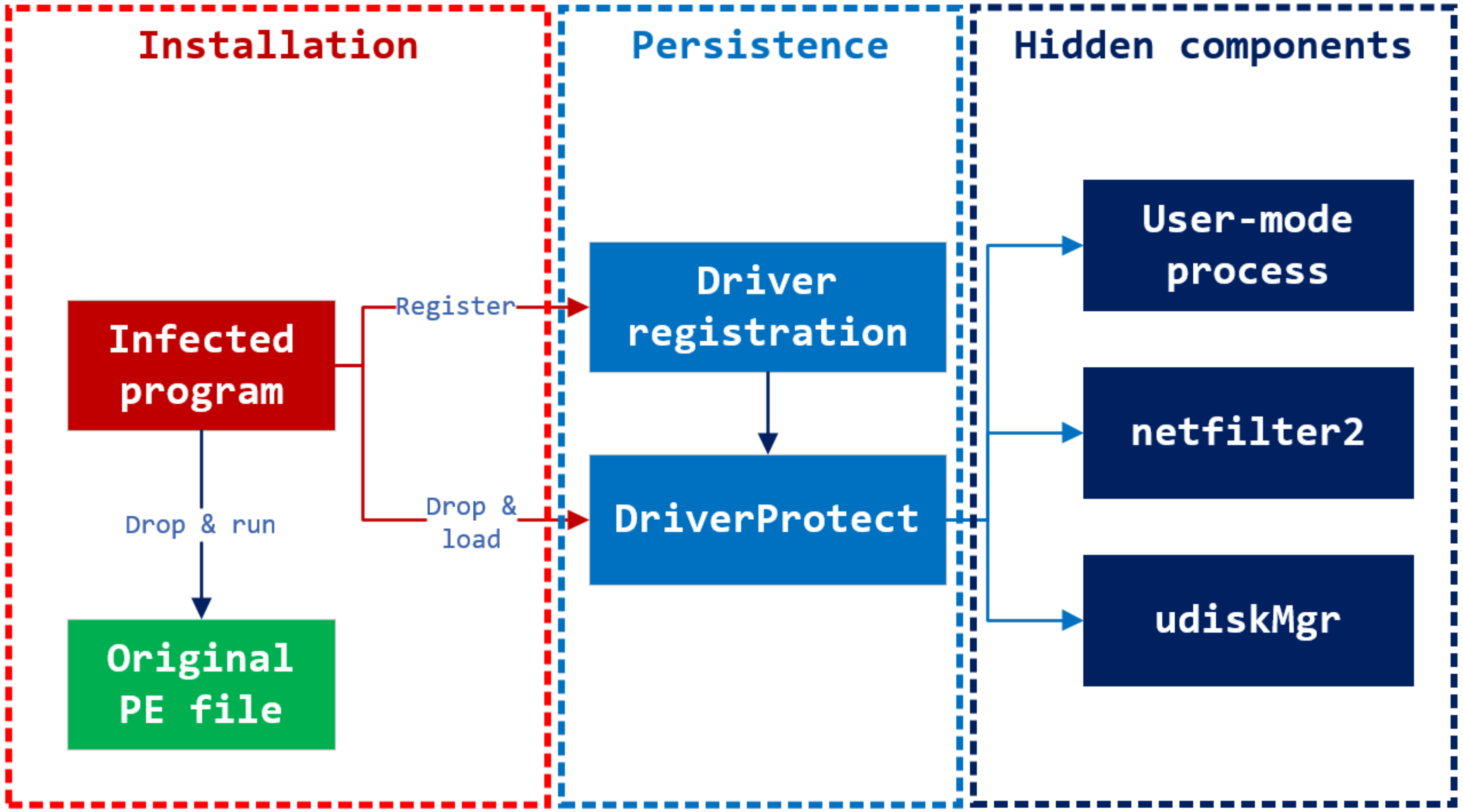
```
000E1D60 1B 54 68 61 77 74 65 20  43 6F 64 65 20 53 69 67  .Thawte Code Sig
000E1D70 6E 69 6E 67 20 43 41 20  2D 20 47 32 02 10 30 10  ning CA - G2..0.
000E1D80 BD B8 26 BA 7F 5E 31 9F  42 41 3D 6D C1 4C 30 09  ½¸&º.^1ŸBA=mÁL0.
000E1D90 06 05 2B 0E 03 02 1A 05  00 A0 70 30 10 06 0A 2B  ..+...... p0...+
000E1DA0 06 01 04 01 82 37 02 01  0C 31 02 30 00 30 19 06  ....‚7...1.0.0..
000E1DB0 09 2A 86 48 86 F7 0D 01  09 03 31 0C 06 0A 2B 06  .*†H†÷....1...+.
000E1DC0 01 04 01 82 37 02 01 04  30 1C 06 0A 2B 06 01 04  ...‚7...0...+...
000E1DD0 01 82 37 02 01 0B 31 0E  30 0C 06 0A 2B 06 01 04  .‚7...1.0...+...
000E1DE0 01 82 37 02 01 16 30 23  06 09 2A 86 48 86 F7 0D  .‚7...0#..*†H†÷.
000E1DF0 01 09 04 31 16 04 14 C9  4F B8 15 F8 F4 6D 49 26  ...1...ÉO¸.øômI&
000E1E00 55 B8 7D 2C 26 18 A2 68  14 D8 94 30 0D 06 09 2A  U¸},&.¢h.Ø”0...*
000E1E10 86 48 86 F7 0D 01 01 01  05 00 04 82 01 00 0C FE  †H†÷.......‚...þ
000E1E20 AE 44 BB 37 D9 DD BA B8  B1 A9 0A 54 05 EB 08 8B  ®D»7ÙÝº¸±©.T.ë.‹
000E1E30 B5 F8 66 9E 93 70 14 36  22 2A A2 D6 4B 33 F4 C7  µøfž“p.6"*¢ÖK3ôÇ
000E1E40 2D 8F AC 36 E4 79 4D F4  6D F7 1B EB A6 32 B9 B7  -.¬6äyMôm÷.ë¦2¹·
000E1E50 5C 34 D8 8D 5F C4 80 C9  F6 C7 78 8A 89 11 C8 42  \4Ø._Ä€ÉöÇxŠ‰.ÈB
000E1E60 75 CD C0 7A F3 8E B9 7F  31 48 9D 85 5C 81 AF 63  uÍÀzóŽ¹.1H.…\.¯c
000E1E70 F9 BE 7F 02 30 A0 E0 BD  52 4A D6 66 80 DE 12 31  ù¾..0 à½RJÖf€Þ.1
000E1E80 46 C6 A4 44 A5 85 CF B0  D1 E8 6C DF 31 46 A4 0F  FÆ¤D¥…Ï°Ñèlß1F¤.
000E1E90 7F 27 FC D4 30 6D BE C9  D7 6B 44 83 76 22 37 CA  .'üÔ0m¾É×kDƒv"7Ê
000E1EA0 EE 3E DC 3C F6 0F D9 AC  DD E8 07 BB A2 B9 24 6A  î>Ü<ö.Ù¬Ýè.»¢¹$j
000E1EB0 03 BC 6C B0 1C CA 6E 72  B0 0C 07 59 C0 0F 8E 8E  .¼l°.Ênr°..YÀ.ŽŽ
000E1EC0 44 B2 F8 B7 86 BE 2F 0E  E1 45 17 25 5F 97 38 08  D²ø·†¾/.áE.%_—8.
000E1ED0 90 3D 10 29 D7 26 AA 6D  B3 8B E5 4D 49 8E 36 1A  .=.)×&ªm³‹åMIŽ6.
000E1EE0 E3 9E C3 14 2A 6D CA BA  1E 2A EA 01 B8 25 AC 83  ãžÃ.*mÊº.*ê.¸%¬ƒ
000E1EF0 9B FE BB AA A6 06 D0 BE  9C D2 12 F2 C7 FA 1A 01  ›þ»ª¦.Ð¾œÒ.òÇú..
000E1F00 96 06 4B 02 6B 6C 80 98  55 6B 98 00 F4 D8 BD 05  –.K.kl€˜Uk˜.ôØ½.
000E1F10 C4 DE 1D 87 DB 36 AA 08  25 21 CB D1 8E 7F 00 00  ÄÞ.‡Û6ª.%!ËÑŽ...
000E1F20 6E 71 74 78 7B 7E 82 85  88 8B 8F 92 95 98 9C 9F  nqtx{~‚…ˆ‹.’•˜œŸ
000E1F30 A2 A5 A9 AC AF B3 B6 B9  BC C0 C3 C6 C9 CD D0 D3  ¢¥©¬¯³¶¹¼ÀÃÆÉÍÐÓ
000E1F40 D6 DA DD E0 E4 E7 EA ED  F1 F4 F7 FA FE 01 04 07  ÖÚÝàäçêíñô÷úþ...
```

000E0E38

Binary differences | Insert | Load time: 0.04 seconds

Persistence

Infection/persistence



Shutdown handler registration

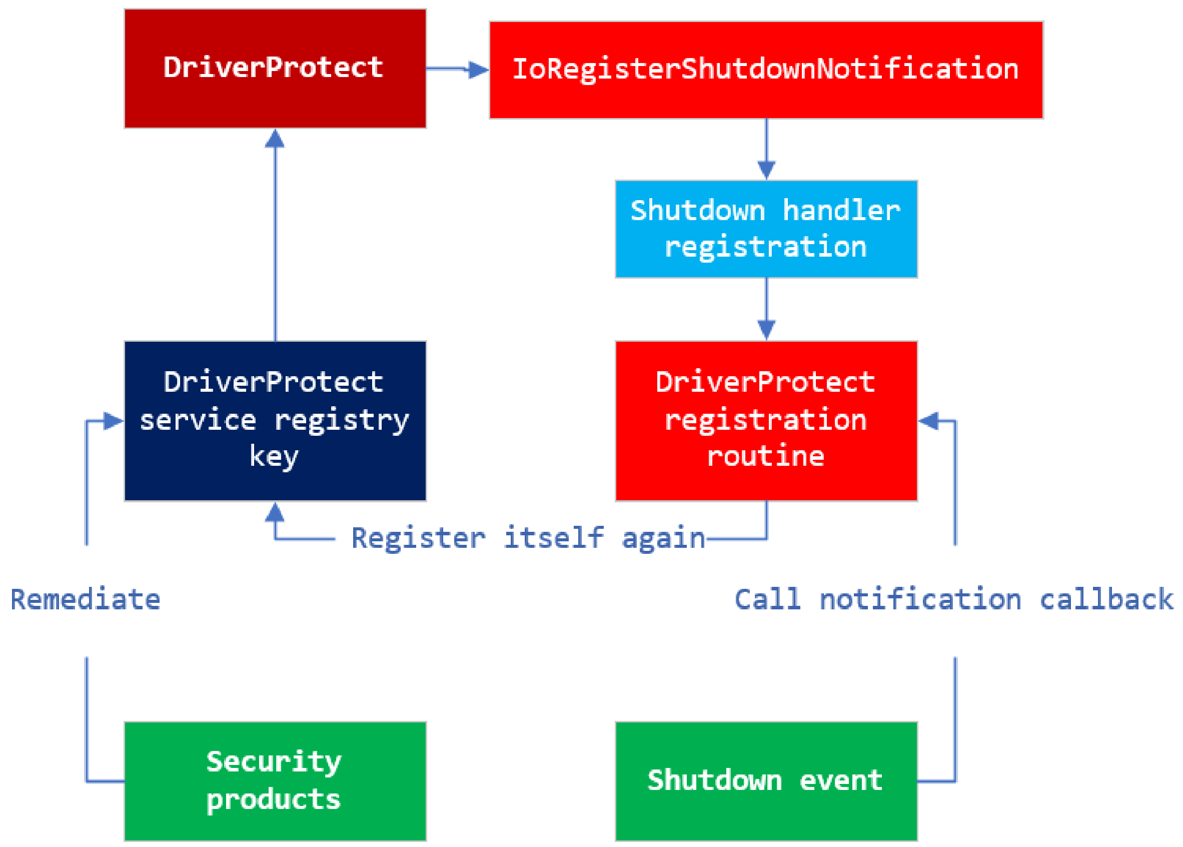
- The *DriverProtect* component will register its main infection routine as a shutdown handler
- With next reboot, the threat will be persistent
- Traditional remediation fails because it doesn’t have control over this handler

Shutdown handler registration

- This routine shows how the shutdown handler is registered
- Remediation is extremely tricky
- The system will be re-infected with each reboot of the system unless the resident kernel driver is unloaded

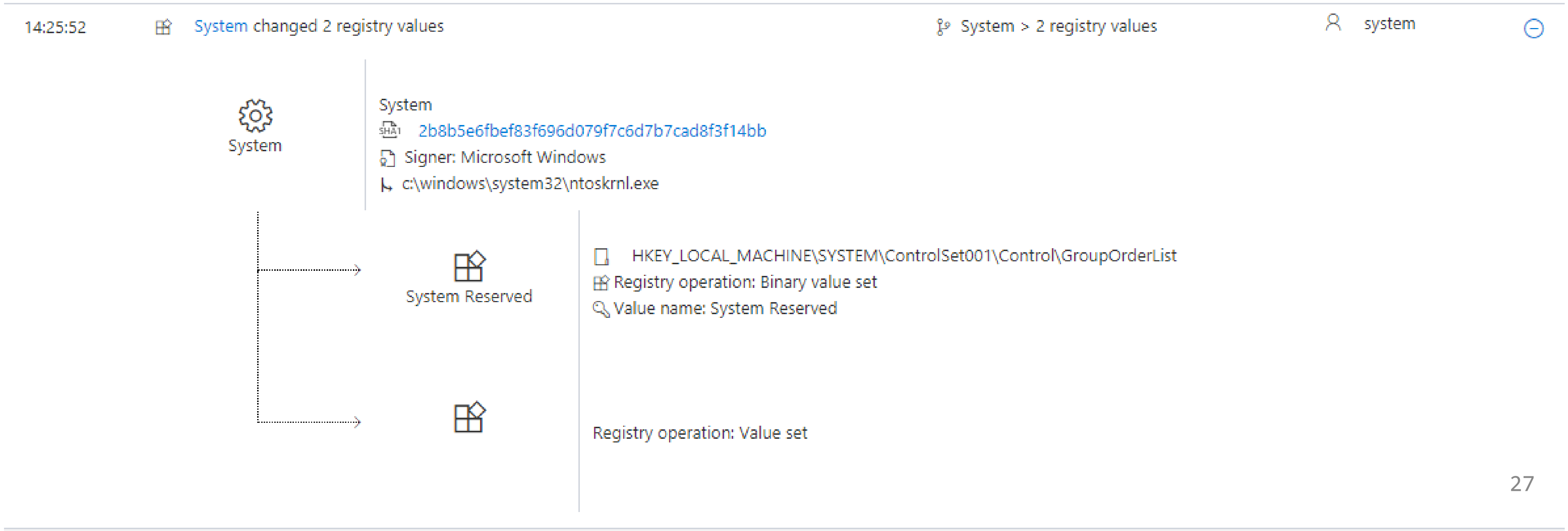
```
IoRegisterShutdownNotification = (void (__fastcall *)(_QWORD))GetSystemRoutineAddress(L"IoRegisterShutdownNotification");
if ( IoRegisterShutdownNotification )
  IoRegisterShutdownNotification(*(_QWORD *)(driverObject + 8));
```

Shutdown handler registration


DriverProtect
IoRegisterShutdownNotification
Shutdown handler registration
DriverProtect service registry key
DriverProtect registration routine
Register itself again
Remediate
Call notification callback
Security products
Shutdown event

Group order list

- The DriverProtect kernel module is in very early stages of the driver loading order
- Will affect following security product related drivers detection attempts



14:25:52 System changed 2 registry values System > 2 registry values system

System

System
2b8b5e6fbef83f696d079f7c6d7b7cad8f3f14bb
Signer: Microsoft Windows
c:\windows\system32\ntoskrnl.exe

System Reserved

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\GroupOrderList
Registry operation: Binary value set
Value name: System Reserved

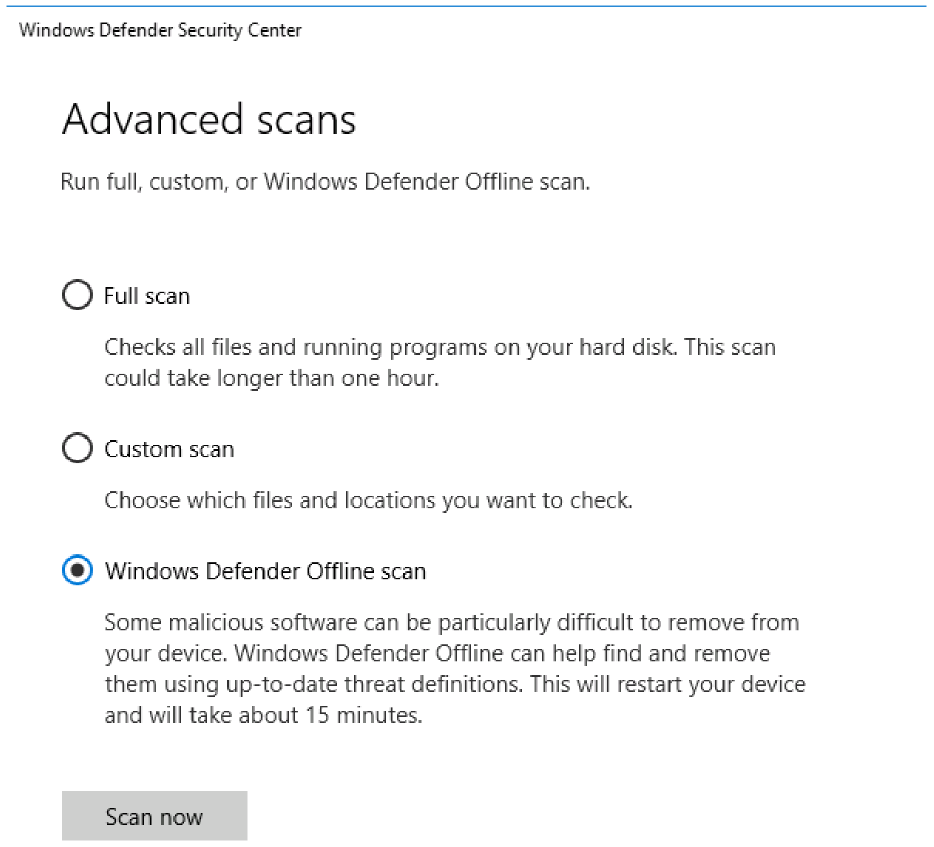
Registry operation: Value set

Windows Defender Offline

- Windows Defender Offline can provide offline remediation capability
- When threat is detected, WDO will:
  - Guide through offline remediation process
  - Cut down the reloading of the rootkit modules

Windows Defender Offline

- WDO will be able to remove the threat
- WDO is a special Defender service where Defender runs scanning from clean OS image from WINRE (Windows Recovery) partition
- Once system reboots, the kernel malicious drivers clean up itself, before any AV scans.



Windows Defender Security Center

Advanced scans

Run full, custom, or Windows Defender Offline scan.

○ Full scan

Checks all files and running programs on your hard disk. This scan could take longer than one hour.

○ Custom scan

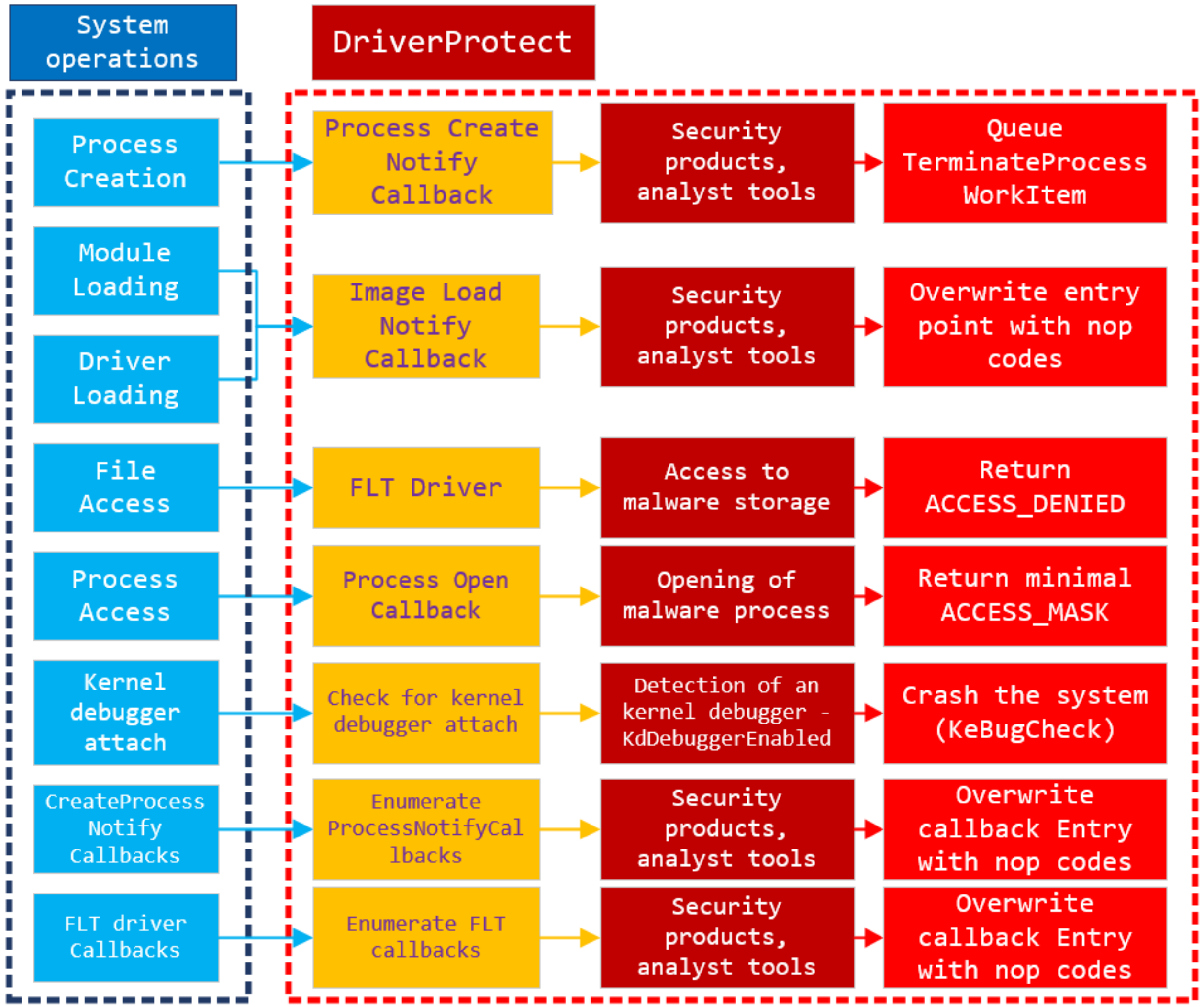
Choose which files and locations you want to check.

[✓ Windows Defender Offline scan]

Some malicious software can be particularly difficult to remove from your device. Windows Defender Offline can help find and remove them using up-to-date threat definitions. This will restart your device and will take about 15 minutes.

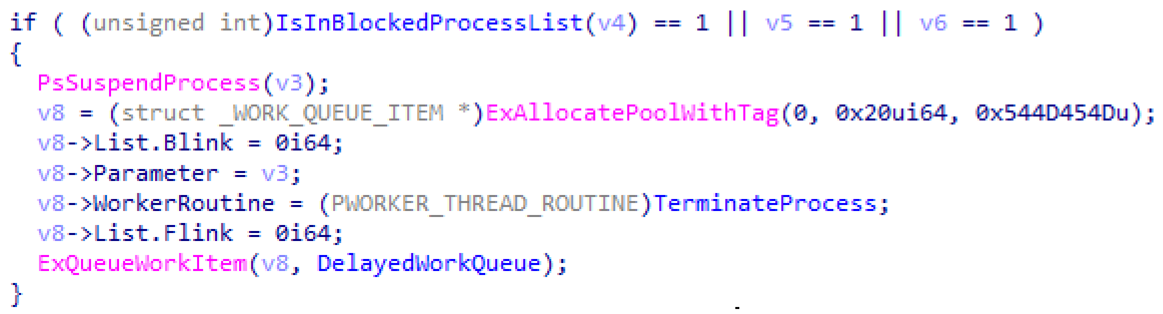
Anti-analysis/debugging

Anti-analysis/debugging

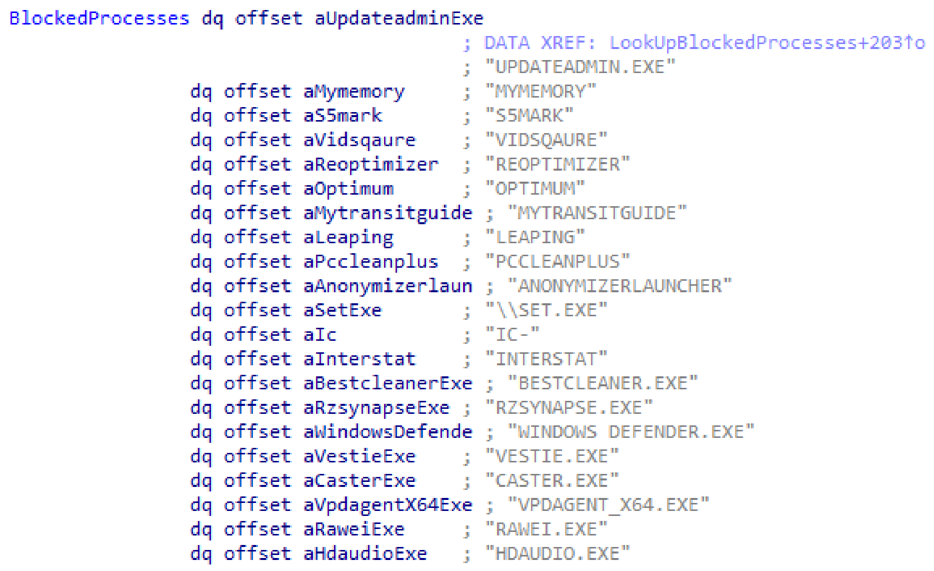

System operations
DriverProtect
Process Creation
Process Create Notify Callback
Security products, analyst tools
Queue TerminateProcess WorkItem
Module Loading
Driver Loading
Image Load Notify Callback
Security products, analyst tools
Overwrite entry point with nop codes
File Access
FLT Driver
Access to malware storage
Return ACCESS_DENIED
Process Access
Process Open Callback
Opening of malware process
Return minimal ACCESS_MASK
Kernel debugger attach
Check for kernel debugger attach
Detection of an kernel debugger - KdDebuggerEnabled
Crash the system (KeBugCheck)
CreateProcess Notify Callbacks
Enumerate ProcessNotifyCallbacks
Security products, analyst tools
Overwrite callback Entry with nop codes
FLT driver Callbacks
Enumerate FLT callbacks
Security products, analyst tools
Overwrite callback Entry with nop codes

Anti-analysis tools

- Process creation callback
  - Also CreateProcess callback is installed by *DriverProtect* to prevent some analysis tools
  - WorkItem queue is used to terminate the process from the callback



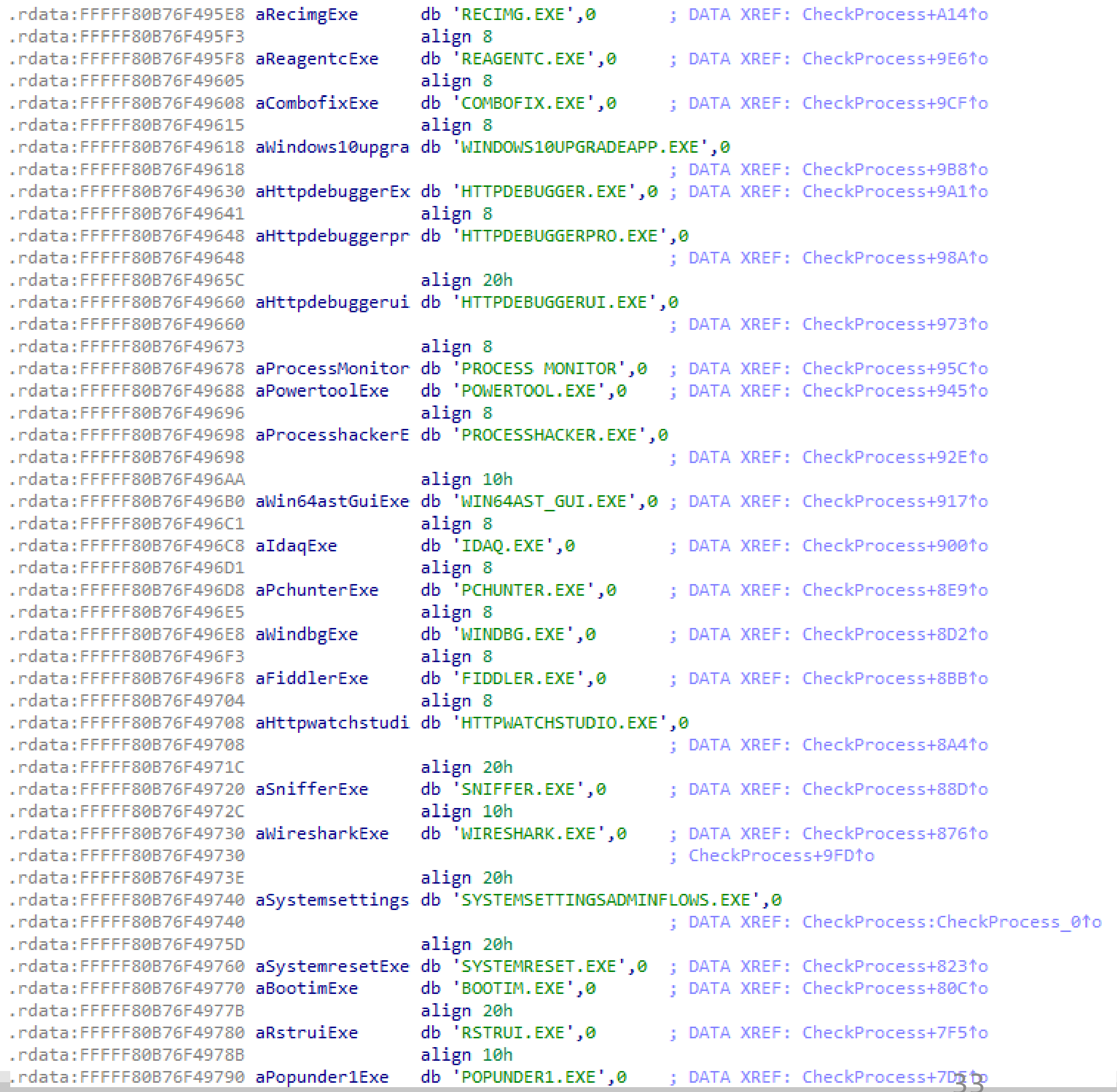
```
if ( (unsigned int)IsInBlockedProcessList(v4) == 1 || v5 == 1 || v6 == 1 )
{
  PsSuspendProcess(v3);
  v8 = (struct _WORK_QUEUE_ITEM *)ExAllocatePoolWithTag(0, 0x20ui64, 0x544D454Du);
  v8->List.Blink = 0i64;
  v8->Parameter = v3;
  v8->WorkerRoutine = (PWORKER_THREAD_ROUTINE)TerminateProcess;
  v8->List.Flink = 0i64;
  ExQueueWorkItem(v8, DelayedWorkQueue);
}
```



```
BlockedProcesses dq offset aUpdateadminExe
                                        ; DATA XREF: LookUpBlockedProcesses+203↑o
                                        ; "UPDATEADMIN.EXE"
                dq offset aMymemory     ; "MYMEMORY"
                dq offset aS5mark       ; "S5MARK"
                dq offset aVidsqaure    ; "VIDSQAURE"
                dq offset aReoptimizer  ; "REOPTIMIZER"
                dq offset aOptimum      ; "OPTIMUM"
                dq offset aMytransitguide ; "MYTRANSITGUIDE"
                dq offset aLeaping      ; "LEAPING"
                dq offset aPccleanplus  ; "PCCLEANPLUS"
                dq offset aAnonymizerlaun ; "ANONYMIZERLAUNCHER"
                dq offset aSetExe       ; "\\SET.EXE"
                dq offset aIc           ; "IC-"
                dq offset aInterstat    ; "INTERSTAT"
                dq offset aBestcleanerExe ; "BESTCLEANER.EXE"
                dq offset aRzsynapseExe ; "RZSYNAPSE.EXE"
                dq offset aWindowsDefende ; "WINDOWS DEFENDER.EXE"
                dq offset aVestieExe    ; "VESTIE.EXE"
                dq offset aCasterExe    ; "CASTER.EXE"
                dq offset aVpdagentX64Exe ; "VPDAGENT_X64.EXE"
                dq offset aRaweiExe     ; "RAWEI.EXE"
                dq offset aHdaudioExe   ; "HDAUDIO.EXE"
```

Anti-analysis tools

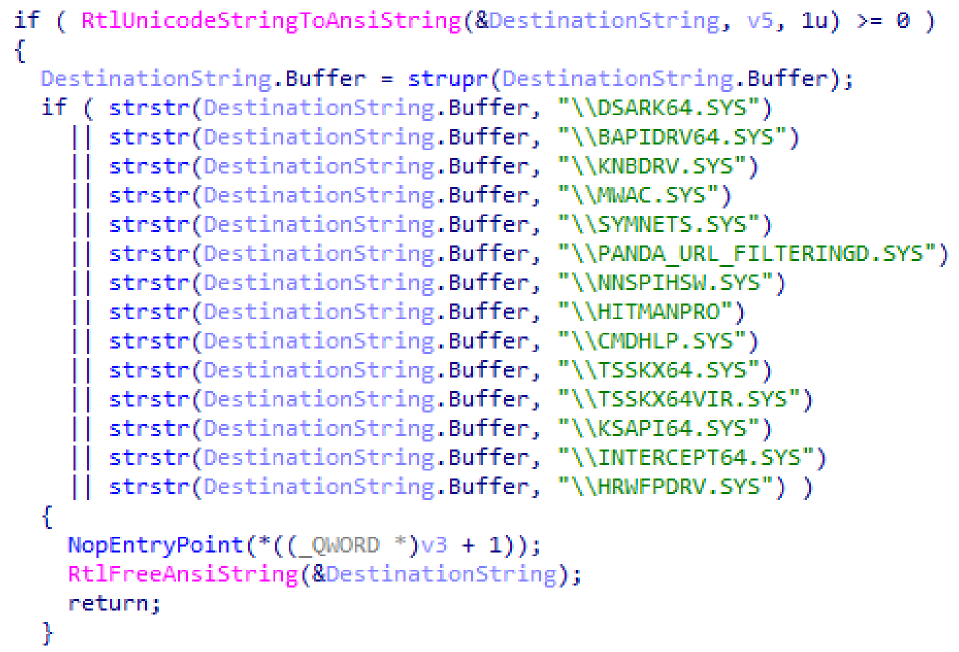
- Image load callback
  - Usual analyst tools are prohibited from launching on the target system from LoadImage callback installed by *DriverProtect* module.



```
.rdata:FFFFF80B76F495E8 aRecimgExe      db 'RECIMG.EXE',0       ; DATA XREF: CheckProcess+A14↑o
.rdata:FFFFF80B76F495F3                 align 8
.rdata:FFFFF80B76F495F8 aReagentcExe    db 'REAGENTC.EXE',0     ; DATA XREF: CheckProcess+9E6↑o
.rdata:FFFFF80B76F49605                 align 8
.rdata:FFFFF80B76F49608 aCombofixExe    db 'COMBOFIX.EXE',0     ; DATA XREF: CheckProcess+9CF↑o
.rdata:FFFFF80B76F49615                 align 8
.rdata:FFFFF80B76F49618 aWindows10upgra db 'WINDOWS10UPGRADEAPP.EXE',0
.rdata:FFFFF80B76F49618                                         ; DATA XREF: CheckProcess+9B8↑o
.rdata:FFFFF80B76F49630 aHttpdebuggerEx db 'HTTPDEBUGGER.EXE',0 ; DATA XREF: CheckProcess+9A1↑o
.rdata:FFFFF80B76F49641                 align 8
.rdata:FFFFF80B76F49648 aHttpdebuggerpr db 'HTTPDEBUGGERPRO.EXE',0
.rdata:FFFFF80B76F49648                                         ; DATA XREF: CheckProcess+98A↑o
.rdata:FFFFF80B76F4965C                 align 20h
.rdata:FFFFF80B76F49660 aHttpdebuggerui db 'HTTPDEBUGGERUI.EXE',0
.rdata:FFFFF80B76F49660                                         ; DATA XREF: CheckProcess+973↑o
.rdata:FFFFF80B76F49673                 align 8
.rdata:FFFFF80B76F49678 aProcessMonitor db 'PROCESS MONITOR',0  ; DATA XREF: CheckProcess+95C↑o
.rdata:FFFFF80B76F49688 aPowertoolExe   db 'POWERTOOL.EXE',0    ; DATA XREF: CheckProcess+945↑o
.rdata:FFFFF80B76F49696                 align 8
.rdata:FFFFF80B76F49698 aProcesshackerE db 'PROCESSHACKER.EXE',0
.rdata:FFFFF80B76F49698                                         ; DATA XREF: CheckProcess+92E↑o
.rdata:FFFFF80B76F496AA                 align 10h
.rdata:FFFFF80B76F496B0 aWin64astGuiExe db 'WIN64AST_GUI.EXE',0 ; DATA XREF: CheckProcess+917↑o
.rdata:FFFFF80B76F496C1                 align 8
.rdata:FFFFF80B76F496C8 aIdaqExe        db 'IDAQ.EXE',0         ; DATA XREF: CheckProcess+900↑o
.rdata:FFFFF80B76F496D1                 align 8
.rdata:FFFFF80B76F496D8 aPchunterExe    db 'PCHUNTER.EXE',0     ; DATA XREF: CheckProcess+8E9↑o
.rdata:FFFFF80B76F496E5                 align 8
.rdata:FFFFF80B76F496E8 aWindbgExe      db 'WINDBG.EXE',0       ; DATA XREF: CheckProcess+8D2↑o
.rdata:FFFFF80B76F496F3                 align 8
.rdata:FFFFF80B76F496F8 aFiddlerExe     db 'FIDDLER.EXE',0      ; DATA XREF: CheckProcess+8BB↑o
.rdata:FFFFF80B76F49704                 align 8
.rdata:FFFFF80B76F49708 aHttpwatchstudi db 'HTTPWATCHSTUDIO.EXE',0
.rdata:FFFFF80B76F49708                                         ; DATA XREF: CheckProcess+8A4↑o
.rdata:FFFFF80B76F4971C                 align 20h
.rdata:FFFFF80B76F49720 aSnifferExe     db 'SNIFFER.EXE',0      ; DATA XREF: CheckProcess+88D↑o
.rdata:FFFFF80B76F4972C                 align 10h
.rdata:FFFFF80B76F49730 aWiresharkExe   db 'WIRESHARK.EXE',0    ; DATA XREF: CheckProcess+876↑o
.rdata:FFFFF80B76F49730                                         ; CheckProcess+9FD↑o
.rdata:FFFFF80B76F4973E                 align 20h
.rdata:FFFFF80B76F49740 aSystemsettings db 'SYSTEMSETTINGSADMINFLOWS.EXE',0
.rdata:FFFFF80B76F49740                                         ; DATA XREF: CheckProcess:CheckProcess_0↑o
.rdata:FFFFF80B76F4975D                 align 20h
.rdata:FFFFF80B76F49760 aSystemresetExe db 'SYSTEMRESET.EXE',0  ; DATA XREF: CheckProcess+823↑o
.rdata:FFFFF80B76F49770 aBootimExe      db 'BOOTIM.EXE',0       ; DATA XREF: CheckProcess+80C↑o
.rdata:FFFFF80B76F4977B                 align 20h
.rdata:FFFFF80B76F49780 aRstruiExe      db 'RSTRUI.EXE',0       ; DATA XREF: CheckProcess+7F5↑o
.rdata:FFFFF80B76F4978B                 align 10h
.rdata:FFFFF80B76F49790 aPopunder1Exe   db 'POPUNDER1.EXE',0    ; DATA XREF: CheckProcess+7D5↑o
```

Anti-detection

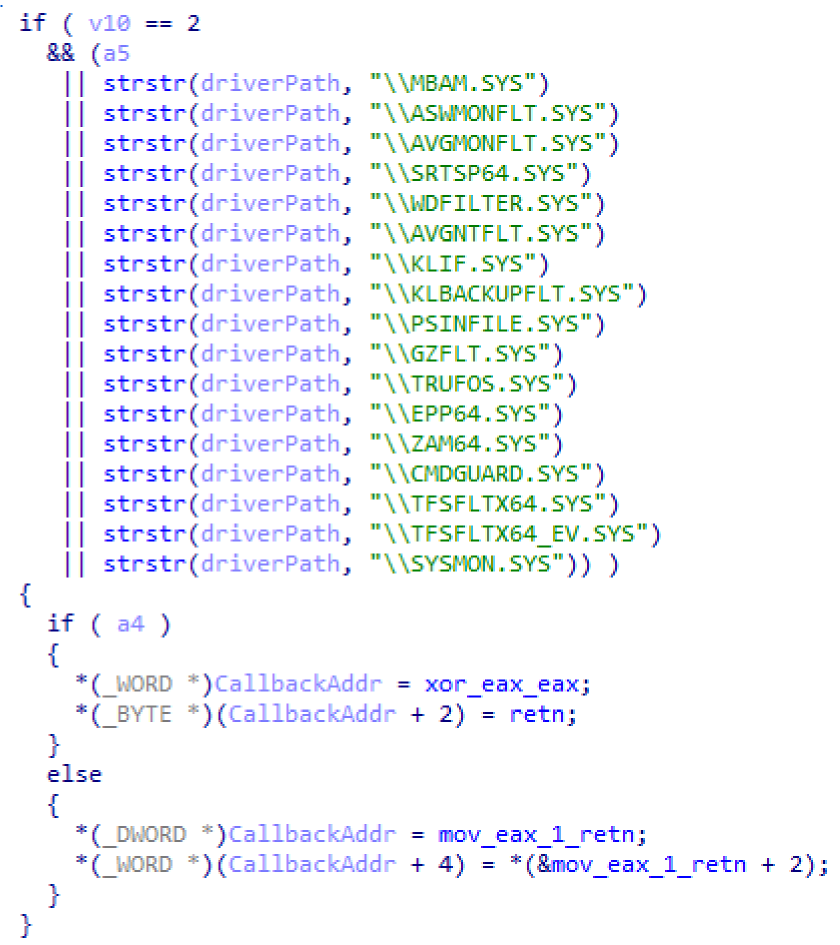
- Monitor Kernel Driver Loading
  - If security product related kernel drivers are loaded, it will nop out the entry point (*xor eax, eax; ret*)



```
if ( RtlUnicodeStringToAnsiString(&DestinationString, v5, 1u) >= 0 )
{
  DestinationString.Buffer = strupr(DestinationString.Buffer);
  if ( strstr(DestinationString.Buffer, "\\DSARK64.SYS")
    || strstr(DestinationString.Buffer, "\\BAPIDRV64.SYS")
    || strstr(DestinationString.Buffer, "\\KNBDRV.SYS")
    || strstr(DestinationString.Buffer, "\\MWAC.SYS")
    || strstr(DestinationString.Buffer, "\\SYMNETS.SYS")
    || strstr(DestinationString.Buffer, "\\PANDA_URL_FILTERINGD.SYS")
    || strstr(DestinationString.Buffer, "\\NNSPIHSW.SYS")
    || strstr(DestinationString.Buffer, "\\HITMANPRO")
    || strstr(DestinationString.Buffer, "\\CMDHLP.SYS")
    || strstr(DestinationString.Buffer, "\\TSSKX64.SYS")
    || strstr(DestinationString.Buffer, "\\TSSKX64VIR.SYS")
    || strstr(DestinationString.Buffer, "\\KSAPI64.SYS")
    || strstr(DestinationString.Buffer, "\\INTERCEPT64.SYS")
    || strstr(DestinationString.Buffer, "\\HRWFPDRV.SYS") )
  {
    NopEntryPoint(*((_QWORD *)v3 + 1));
    RtlFreeAnsiString(&DestinationString);
    return;
  }
```

Anti-detection

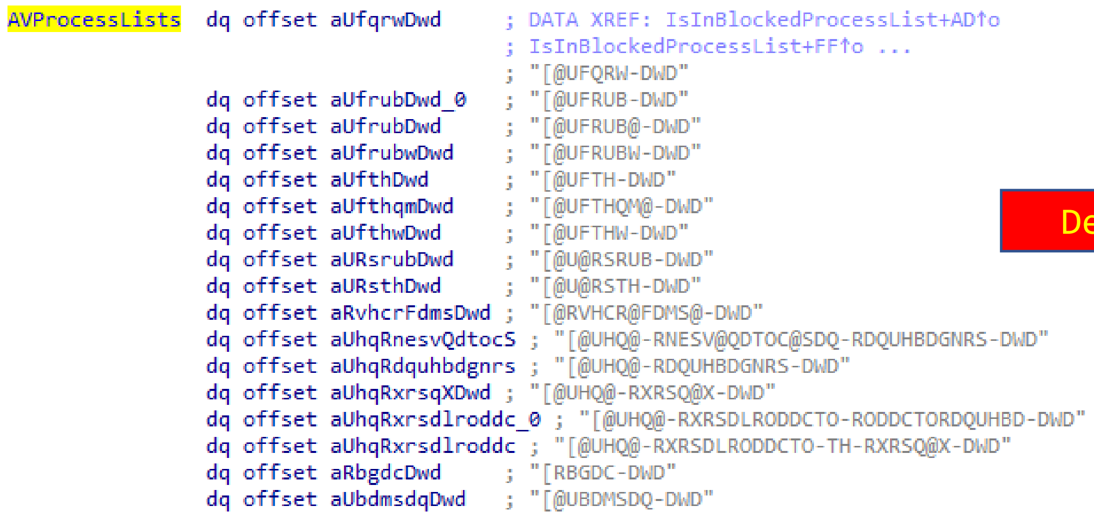
- Disable *ProcessCreateCallbacks* and FLT callbacks
  - The anti-analysis code will enumerate *PspCreateProcessNotifyCallback* and FLT driver routines
  - If it is registered by a security products, it will put nop return instructions over the callback
  - The determination logic for security products include driver path comparison and driver PE header scanning for version information



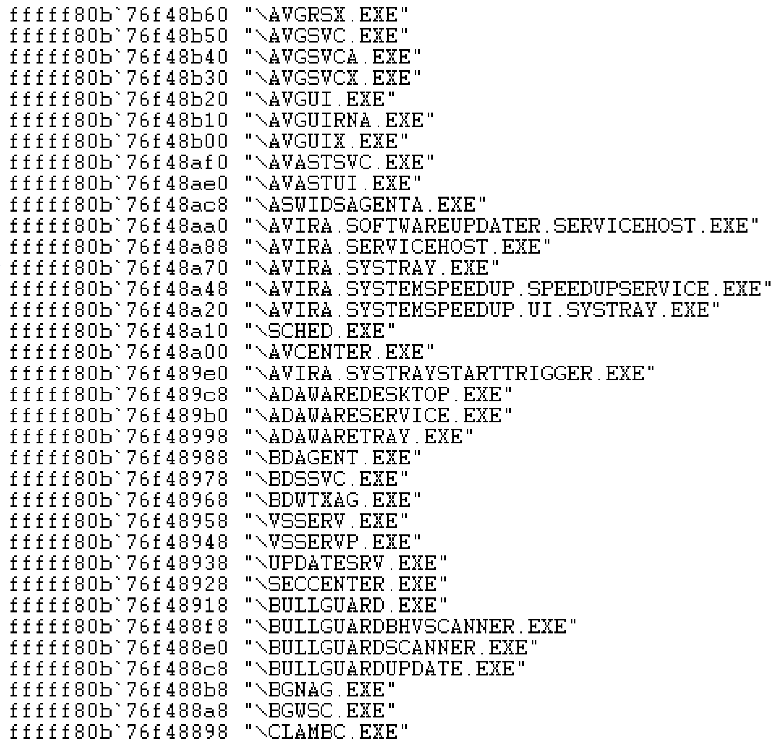
```
if ( v10 == 2
  && (a5
   || strstr(driverPath, "\\MBAM.SYS")
   || strstr(driverPath, "\\ASWMONFLT.SYS")
   || strstr(driverPath, "\\AVGMONFLT.SYS")
   || strstr(driverPath, "\\SRTSP64.SYS")
   || strstr(driverPath, "\\WDFILTER.SYS")
   || strstr(driverPath, "\\AVGNTFLT.SYS")
   || strstr(driverPath, "\\KLIF.SYS")
   || strstr(driverPath, "\\KLBACKUPFLT.SYS")
   || strstr(driverPath, "\\PSINFILE.SYS")
   || strstr(driverPath, "\\GZFLT.SYS")
   || strstr(driverPath, "\\TRUFOS.SYS")
   || strstr(driverPath, "\\EPP64.SYS")
   || strstr(driverPath, "\\ZAM64.SYS")
   || strstr(driverPath, "\\CMDGUARD.SYS")
   || strstr(driverPath, "\\TFSFLTX64.SYS")
   || strstr(driverPath, "\\TFSFLTX64_EV.SYS")
   || strstr(driverPath, "\\SYSMON.SYS")) )
{
  if ( a4 )
  {
    *(_WORD *)CallbackAddr = xor_eax_eax;
    *(_BYTE *)(CallbackAddr + 2) = retn;
  }
  else
  {
    *(_DWORD *)CallbackAddr = mov_eax_1_retn;
    *(_WORD *)(CallbackAddr + 4) = *(&mov_eax_1_retn + 2);
  }
}
```

Anti-detection: Security products

- The *DriverProtect* has extensive list of Anti-malware product processes. They are encoded in file and decoded dynamically.



```
AVProcessLists  dq offset aUfqrwDwd     ; DATA XREF: IsInBlockedProcessList+AD↑o
                                        ; IsInBlockedProcessList+FF↑o ...
                                        ; "[@UFQRW-DWD"
                dq offset aUfrubDwd_0   ; "[@UFRUB-DWD"
                dq offset aUfrubDwd     ; "[@UFRUB@-DWD"
                dq offset aUfrubwDwd    ; "[@UFRUBW-DWD"
                dq offset aUfthDwd      ; "[@UFTH-DWD"
                dq offset aUfthqmDwd    ; "[@UFTHQM@-DWD"
                dq offset aUfthwDwd     ; "[@UFTHW-DWD"
                dq offset aURsrubDwd    ; "[@U@RSRUB-DWD"
                dq offset aURsthDwd     ; "[@U@RSTH-DWD"
                dq offset aRvhcrFdmsDwd ; "[@RVHCR@FDMS@-DWD"
                dq offset aUhqRnesvQdtocS ; "[@UHQ@-RNESV@QDTOC@SDQ-RDQUHBDGNRS-DWD"
                dq offset aUhqRdquhbdgnrs ; "[@UHQ@-RDQUHBDGNRS-DWD"
                dq offset aUhqRxrsqXDwd ; "[@UHQ@-RXRSQ@X-DWD"
                dq offset aUhqRxrsdlroddc_0 ; "[@UHQ@-RXRSDLRODDCTO-RODDCTORDQUHBD-DWD"
                dq offset aUhqRxrsdlroddc ; "[@UHQ@-RXRSDLRODDCTO-TH-RXRSQ@X-DWD"
                dq offset aRbgdcDwd     ; "[RBGDC-DWD"
                dq offset aUbdmsdqDwd   ; "[@UBDMSDQ-DWD"
```



```
fffff80b`76f48b60  "\AVGRSX.EXE"
fffff80b`76f48b50  "\AVGSVC.EXE"
fffff80b`76f48b40  "\AVGSVCA.EXE"
fffff80b`76f48b30  "\AVGSVCX.EXE"
fffff80b`76f48b20  "\AVGUI.EXE"
fffff80b`76f48b10  "\AVGUIRNA.EXE"
fffff80b`76f48b00  "\AVGUIX.EXE"
fffff80b`76f48af0  "\AVASTSVC.EXE"
fffff80b`76f48ae0  "\AVASTUI.EXE"
fffff80b`76f48ac8  "\ASWIDSAGENTA.EXE"
fffff80b`76f48aa0  "\AVIRA.SOFTWAREUPDATER.SERVICEHOST.EXE"
fffff80b`76f48a88  "\AVIRA.SERVICEHOST.EXE"
fffff80b`76f48a70  "\AVIRA.SYSTRAY.EXE"
fffff80b`76f48a48  "\AVIRA.SYSTEMSPEEDUP.SPEEDUPSERVICE.EXE"
fffff80b`76f48a20  "\AVIRA.SYSTEMSPEEDUP.UI.SYSTRAY.EXE"
fffff80b`76f48a10  "\SCHED.EXE"
fffff80b`76f48a00  "\AVCENTER.EXE"
fffff80b`76f489e0  "\AVIRA.SYSTRAYSTARTTRIGGER.EXE"
fffff80b`76f489c8  "\ADAWAREDESKTOP.EXE"
fffff80b`76f489b0  "\ADAWARESERVICE.EXE"
fffff80b`76f48998  "\ADAWARETRAY.EXE"
fffff80b`76f48988  "\BDAGENT.EXE"
fffff80b`76f48978  "\BDSSVC.EXE"
fffff80b`76f48968  "\BDWTXAG.EXE"
fffff80b`76f48958  "\VSSERV.EXE"
fffff80b`76f48948  "\VSSERVP.EXE"
fffff80b`76f48938  "\UPDATESRV.EXE"
fffff80b`76f48928  "\SECCENTER.EXE"
fffff80b`76f48918  "\BULLGUARD.EXE"
fffff80b`76f488f8  "\BULLGUARDBHVSCANNER.EXE"
fffff80b`76f488e0  "\BULLGUARDSCANNER.EXE"
fffff80b`76f488c8  "\BULLGUARDUPDATE.EXE"
fffff80b`76f488b8  "\BGNAG.EXE"
fffff80b`76f488a8  "\BGWSC.EXE"
fffff80b`76f48898  "\CLAMBC.EXE"
```

Anti-detection: Security products

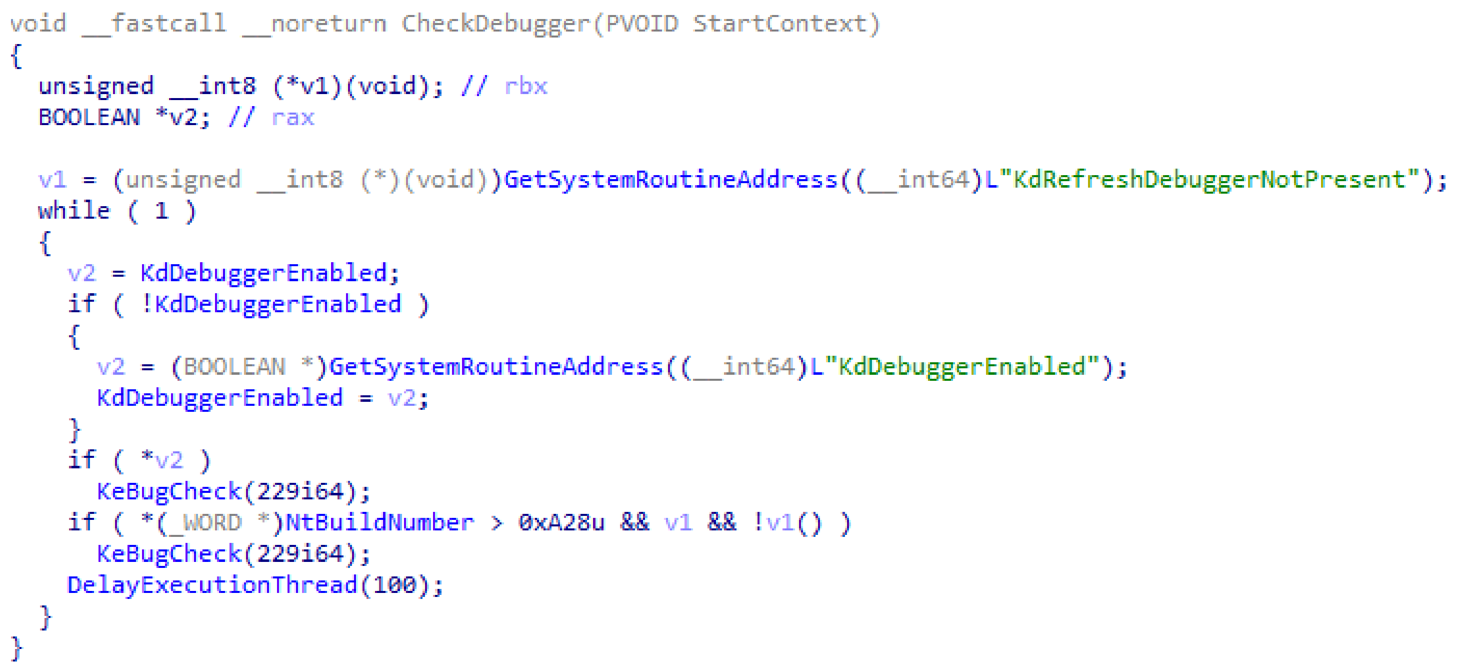
- Scanning happens upon PE header
  - *ReadFile -> ScanSecurityProductPatterns*

```
fileBuffer = (signed int *)ReadFile(&UnicodeString, &v16);
if ( fileBuffer )
{
  v10 = KeAcquireSpinLockRaiseToDpc(&SpinLock);
  v11 = a6;
  v12 = v10;
  if ( a6 )
    v13 = ScanSecurityProductPatterns(fileBuffer, 0i64, v16, 2);
  else
    v13 = ScanSecurityProductPatterns(fileBuffer, 0i64, v16, 1);
  KeReleaseSpinLock(&SpinLock, v12);
  if ( v13 )
    NopPspCreateNotifyCallback(driverPathUpr, EntryPointAddr, v6, a5, v11);
```

```
2: kd> dqa FFFFF80B76F4E020 L50
fffff80b`76f4e020  fffff80b`76f49564 "AVG"
fffff80b`76f4e028  fffff80b`76f4955c "AVAST"
fffff80b`76f4e030  fffff80b`76f49554 "Avira"
fffff80b`76f4e038  fffff80b`76f49548 "Lavasoft"
fffff80b`76f4e040  fffff80b`76f4953c "AhnLab"
fffff80b`76f4e048  fffff80b`76f49530 "Bitdefender"
fffff80b`76f4e050  fffff80b`76f49520 "BullGuard"
fffff80b`76f4e058  fffff80b`76f49518 "Immunet"
fffff80b`76f4e060  fffff80b`76f49508 "Emsisoft"
fffff80b`76f4e068  fffff80b`76f494fc "ESET"
fffff80b`76f4e070  fffff80b`76f494f0 "Kaspersky"
fffff80b`76f4e078  fffff80b`76f494e0 "Malwarebytes"
fffff80b`76f4e080  fffff80b`76f494d8 "McAfee"
fffff80b`76f4e088  fffff80b`76f494c8 "Panda Security"

1: kd> dqa FFFFF80B76F4E210
fffff80b`76f4e210  fffff80b`76f49148 "AVG Technologies CZ, s.r.o."
fffff80b`76f4e218  fffff80b`76f49130 "AVAST Software a.s."
fffff80b`76f4e220  fffff80b`76f49118 "AVAST Software s.r.o."
fffff80b`76f4e228  fffff80b`76f490f8 "Avira Operations GmbH & Co.KG"
fffff80b`76f4e230  fffff80b`76f490e0 "Lavasoft Limited"
fffff80b`76f4e238  fffff80b`76f490d0 "Bitdefender SRL"
fffff80b`76f4e240  fffff80b`76f490c0 "BullGuard Ltd"
fffff80b`76f4e248  fffff80b`76f490b0 "BullGuard Ltd."
fffff80b`76f4e250  fffff80b`76f49098 "Immunet Corporation"
fffff80b`76f4e258  fffff80b`76f49088 "Emsisoft GmbH"
fffff80b`76f4e260  fffff80b`76f49070 "ESET, spol.s r.o."
fffff80b`76f4e268  fffff80b`76f49060 "Kaspersky Lab"
fffff80b`76f4e270  fffff80b`76f49040 "Malwarebytes Corporation"
fffff80b`76f4e278  fffff80b`76f49030 "McAfee, Inc."
fffff80b`76f4e280  fffff80b`76f49018 "Panda Security S.L"
fffff80b`76f4e288  fffff80b`76f49000 "Blue Coat Norway AS"
```

Anti-debugging: Kernel debugger check

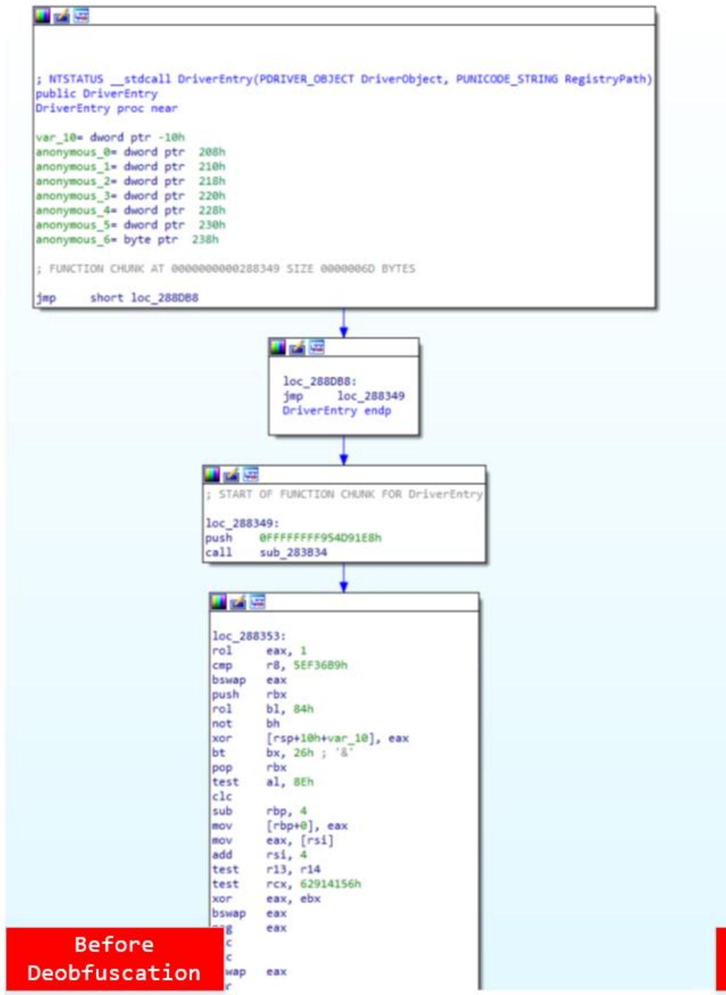
- The rootkit checks whether kernel debugger is enabled
- If enabled, it will call *KeBugCheck*.



```
void __fastcall __noreturn CheckDebugger(PVOID StartContext)
{
  unsigned __int8 (*v1)(void); // rbx
  BOOLEAN *v2; // rax

  v1 = (unsigned __int8 (*)(void))GetSystemRoutineAddress((__int64)L"KdRefreshDebuggerNotPresent");
  while ( 1 )
  {
    v2 = KdDebuggerEnabled;
    if ( !KdDebuggerEnabled )
    {
      v2 = (BOOLEAN *)GetSystemRoutineAddress((__int64)L"KdDebuggerEnabled");
      KdDebuggerEnabled = v2;
    }
    if ( *v2 )
      KeBugCheck(229i64);
    if ( *(_WORD *)NtBuildNumber > 0xA28u && v1 && !v1() )
      KeBugCheck(229i64);
    DelayExecutionThread(100);
  }
}
```

Anti-analysis: obfuscations

- Some rootkit kernel images are obfuscated in file (VMProtect)
- When it is loaded in the kernel, it will unpack itself with original contents
- It will not create new +RWX kernel memory, but will use existing section memory to de-obfuscate itself

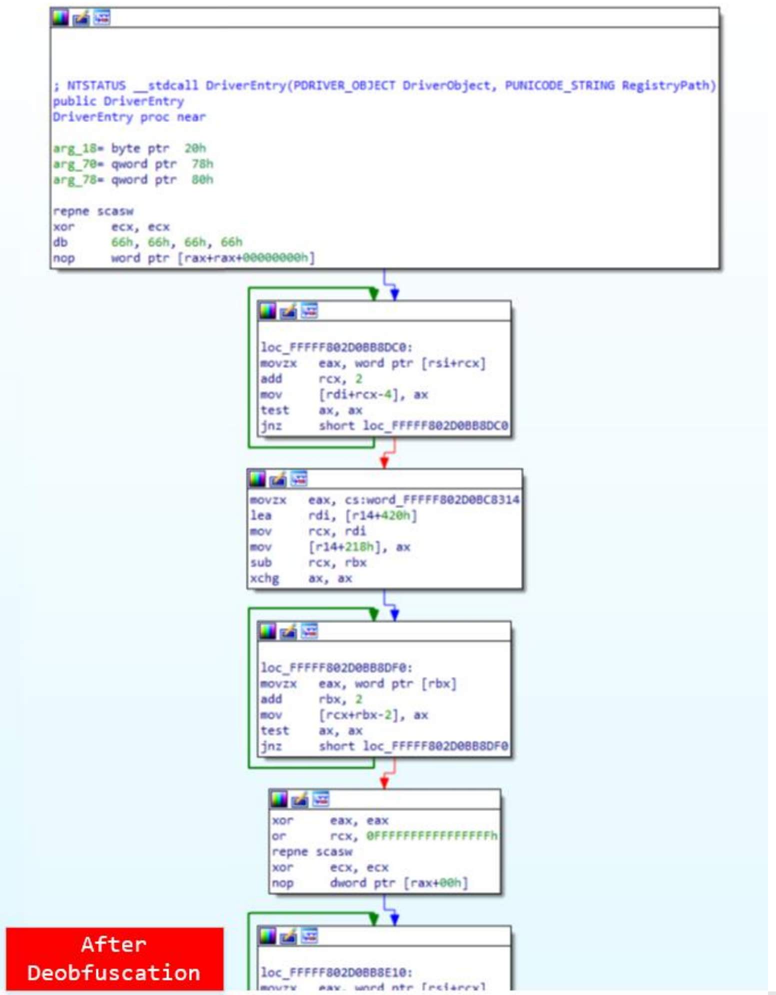


```
; NTSTATUS __stdcall DriverEntry(PDRIVER_OBJECT DriverObject, PUNICODE_STRING RegistryPath)
public DriverEntry
DriverEntry proc near

var_10= dword ptr -10h
anonymous_0= dword ptr  208h
anonymous_1= dword ptr  210h
anonymous_2= dword ptr  218h
anonymous_3= dword ptr  220h
anonymous_4= dword ptr  228h
anonymous_5= dword ptr  230h
anonymous_6= byte ptr  238h

; FUNCTION CHUNK AT 0000000000288349 SIZE 0000006D BYTES

jmp     short loc_288DB8
```

```
loc_288DB8:
jmp     loc_288349
DriverEntry endp
```

```
; START OF FUNCTION CHUNK FOR DriverEntry

loc_288349:
push    0FFFFFFFF954D91E8h
call    sub_283834
```

```
loc_288353:
rol     eax, 1
cmp     r8, 5EF3689h
bswap   eax
push    rbx
rol     bl, 84h
not     bh
xor     [rsp+10h+var_10], eax
bt      bx, 26h ; '&'
pop     rbx
test    al, 8Eh
clc
sub     rbp, 4
mov     [rbp+0], eax
mov     eax, [rsi]
add     rsi, 4
test    r13, r14
test    rcx, 62914156h
xor     eax, ebx
bswap   eax
g       eax
c
c
wap     eax
```



```
; NTSTATUS __stdcall DriverEntry(PDRIVER_OBJECT DriverObject, PUNICODE_STRING RegistryPath)
public DriverEntry
DriverEntry proc near

arg_18= byte ptr  20h
arg_70= qword ptr  78h
arg_78= qword ptr  80h

repne scasw
xor     ecx, ecx
db      66h, 66h, 66h, 66h
nop     word ptr [rax+rax+00000000h]
```

```
loc_FFFFF802D0BB8DC0:
movzx   eax, word ptr [rsi+rcx]
add     rcx, 2
mov     [rdi+rcx-4], ax
test    ax, ax
jnz     short loc_FFFFF802D0BB8DC0
```

```
movzx   eax, cs:word_FFFFF802D0BC8314
lea     rdi, [r14+420h]
mov     rcx, rdi
mov     [r14+218h], ax
sub     rcx, rbx
xchg    ax, ax
```

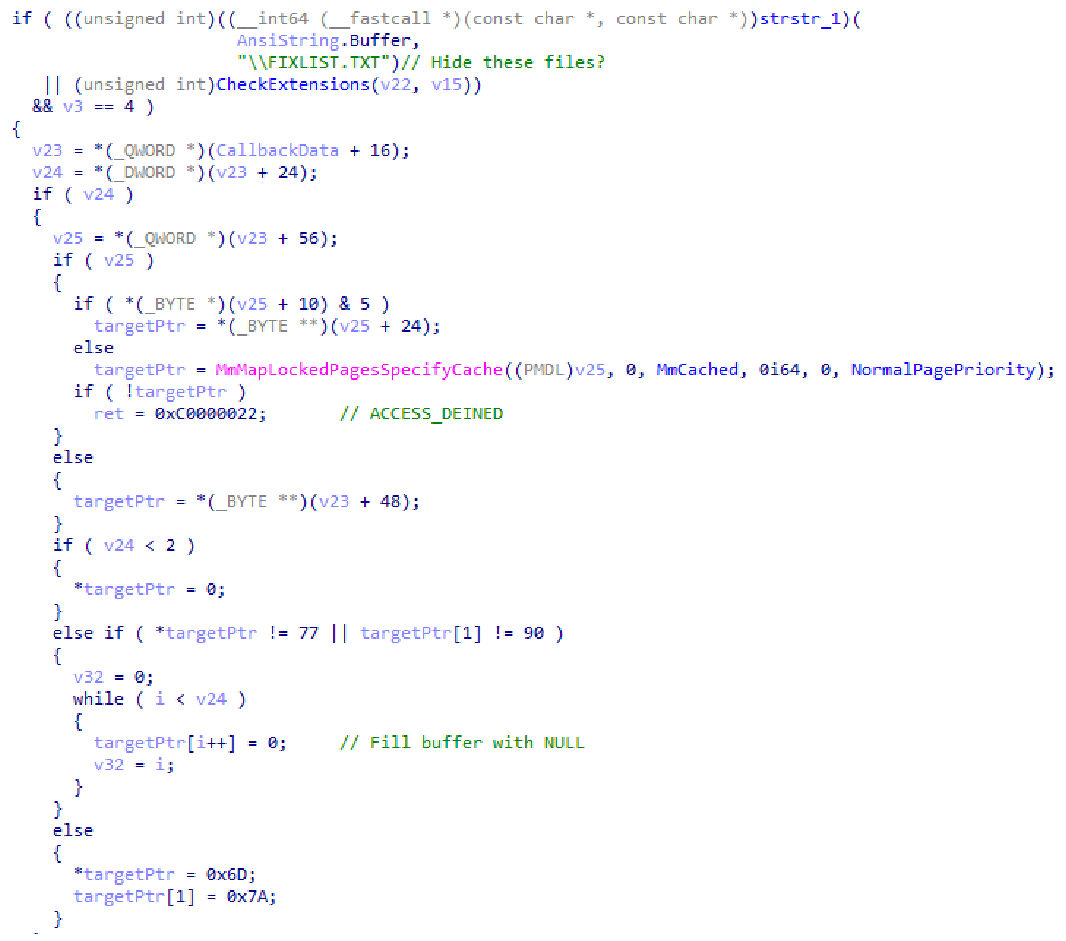
```
loc_FFFFF802D0BB8DF0:
movzx   eax, word ptr [rbx]
add     rbx, 2
mov     [rcx+rbx-2], ax
test    ax, ax
jnz     short loc_FFFFF802D0BB8DF0
```

```
xor     eax, eax
or      rcx, 0FFFFFFFFFFFFFFFFh
repne scasw
xor     ecx, ecx
nop     dword ptr [rax+00h]
```

```
loc_FFFFF802D0BB8E10:
```

Interfering with recovery tool

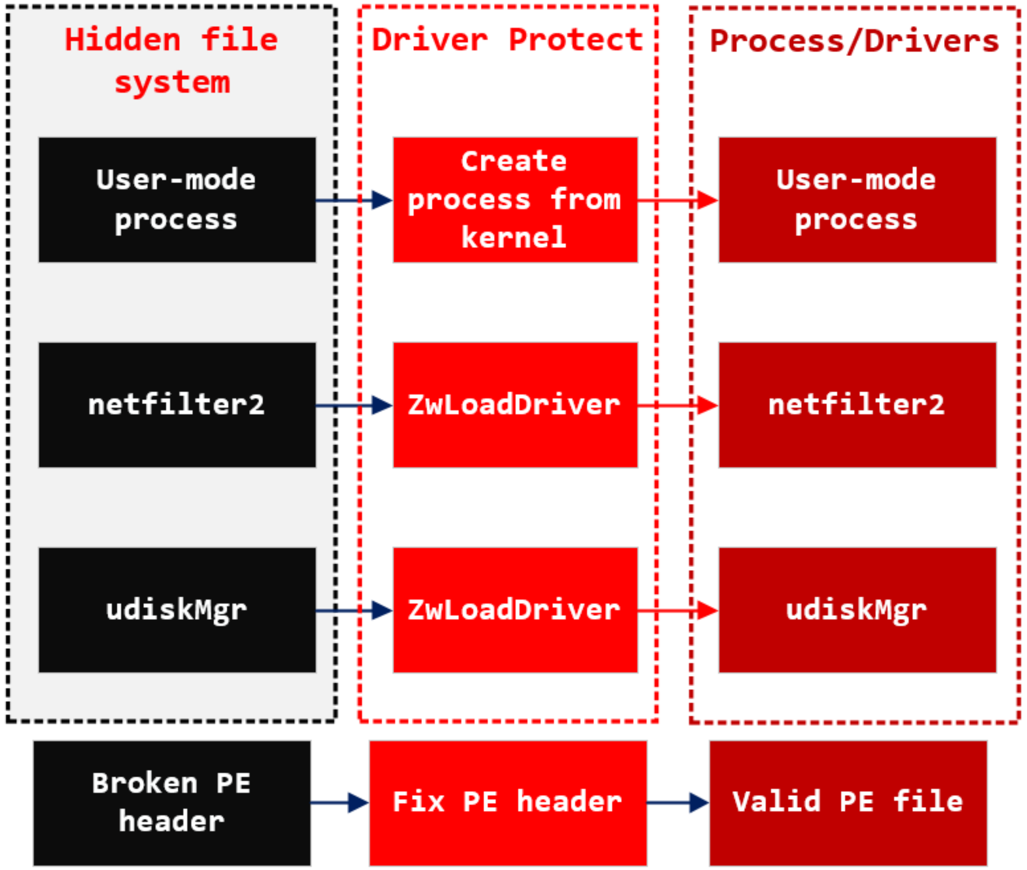
- If any process tries to write contents to FIXLIST.TXT, the contents will be replaced with NULLs.
- FIXLIST.TXT is used by Farbar Recovery Scan Tool



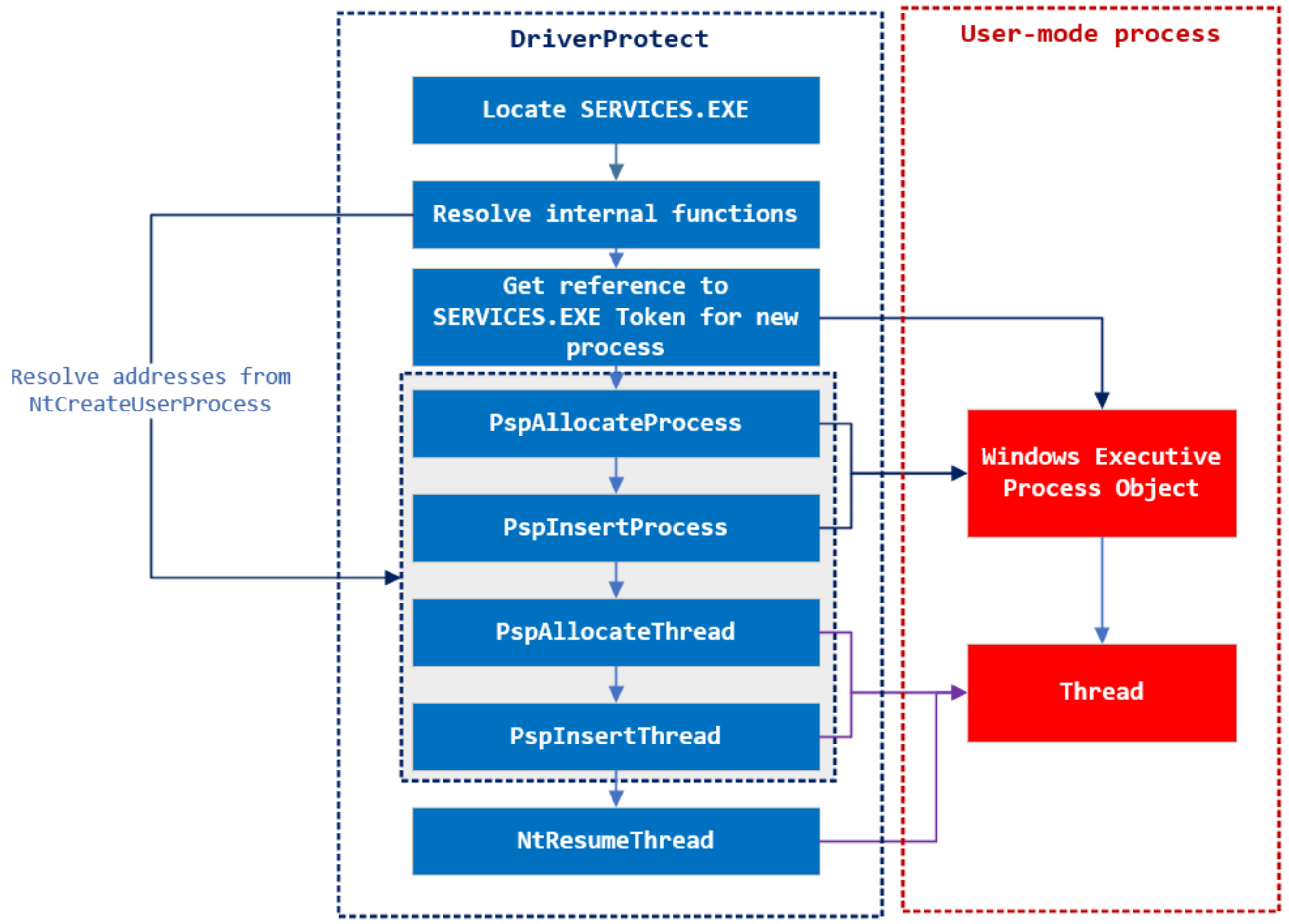
```
if ( ((unsigned int)((__int64 (__fastcall *)(const char *, const char *))strstr_1)(
                      AnsiString.Buffer,
                      "\\FIXLIST.TXT")// Hide these files?
   || (unsigned int)CheckExtensions(v22, v15))
  && v3 == 4 )
{
  v23 = *(_QWORD *)(CallbackData + 16);
  v24 = *(_DWORD *)(v23 + 24);
  if ( v24 )
  {
    v25 = *(_QWORD *)(v23 + 56);
    if ( v25 )
    {
      if ( *(_BYTE *)(v25 + 10) & 5 )
        targetPtr = *(_BYTE **)(v25 + 24);
      else
        targetPtr = MmMapLockedPagesSpecifyCache((PMDL)v25, 0, MmCached, 0i64, 0, NormalPagePriority);
      if ( !targetPtr )
        ret = 0xC0000022;       // ACCESS_DEINED
    }
    else
    {
      targetPtr = *(_BYTE **)(v23 + 48);
    }
    if ( v24 < 2 )
    {
      *targetPtr = 0;
    }
    else if ( *targetPtr != 77 || targetPtr[1] != 90 )
    {
      v32 = 0;
      while ( i < v24 )
      {
        targetPtr[i++] = 0;     // Fill buffer with NULL
        v32 = i;
      }
    }
    else
    {
      *targetPtr = 0x6D;
      targetPtr[1] = 0x7A;
    }
```

Hidden file system

Hidden file system – loading components



Creating user-mode process

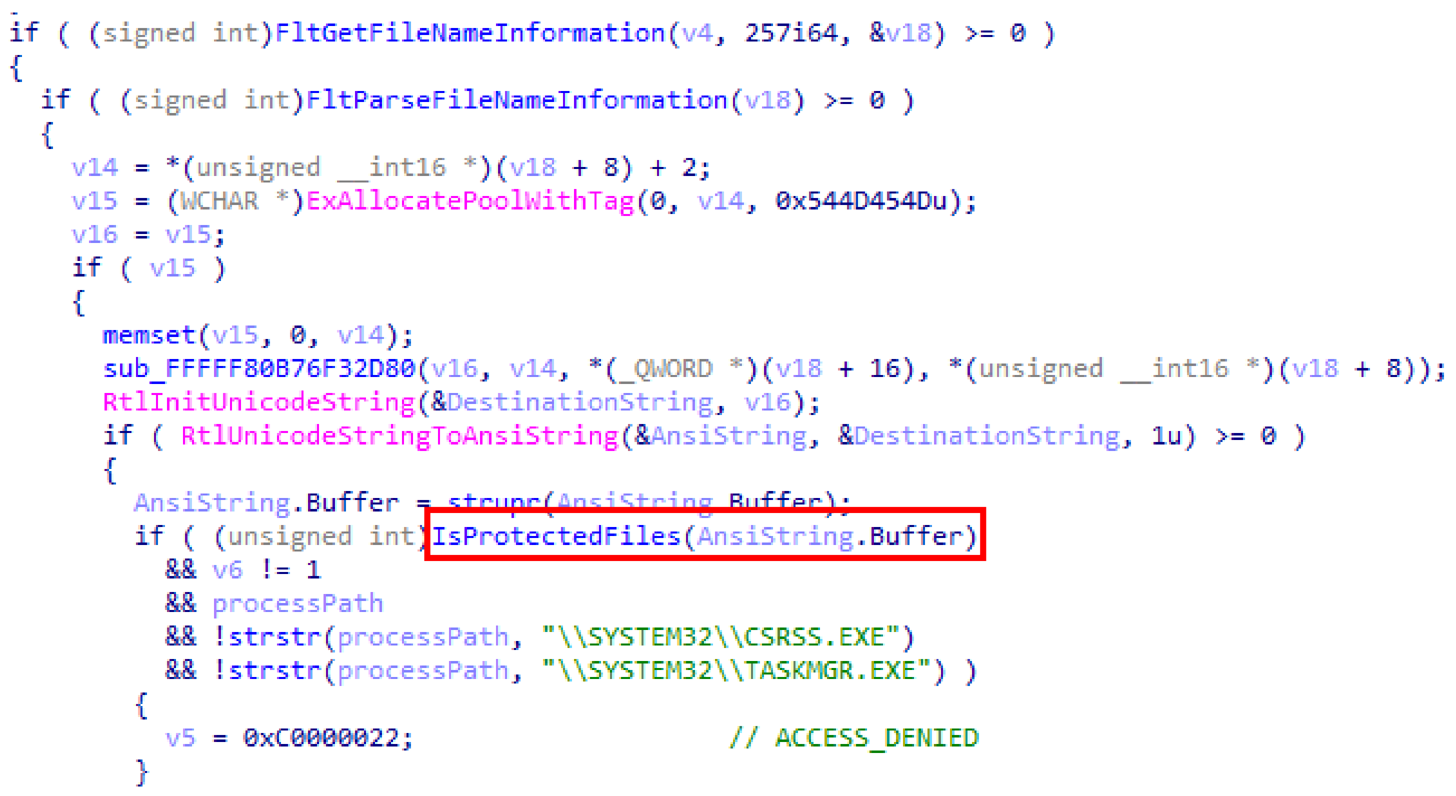

DriverProtect
User-mode process
Locate SERVICES.EXE
Resolve internal functions
Get reference to SERVICES.EXE Token for new process
Resolve addresses from NtCreateUserProcess
PspAllocateProcess
PspInsertProcess
PspAllocateThread
PspInsertThread
NtResumeThread
Windows Executive Process Object
Thread

Hidden file system

- The *DriverProtect* module installs filter driver module to limit access to it’s components
- The other core drivers (*netfilter2* and *udiskMgr*) will be directly loaded from the *DriverProtect* itself
- Security products’ operation in user-mode will have limited visibility into the core files because they are loaded from hidden file location

Hidden file system

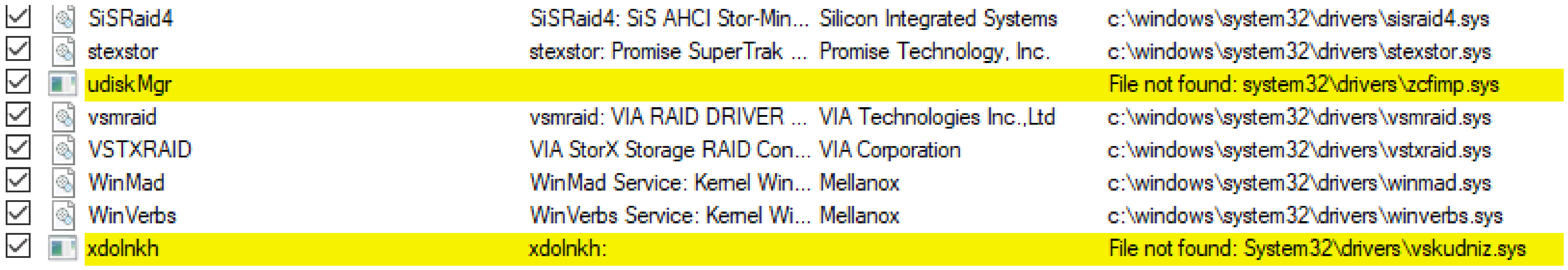
- *DriverProtect* will filter out access to protected files (malware components).
- Security products run in userspace will fail to access malware components
- The file contents inside protected storage is broken PE and will patched up when loaded into memory
- It is implemented as filtering driver



```
if ( (signed int)FltGetFileNameInformation(v4, 257i64, &v18) >= 0 )
{
  if ( (signed int)FltParseFileNameInformation(v18) >= 0 )
  {
    v14 = *(unsigned __int16 *)(v18 + 8) + 2;
    v15 = (WCHAR *)ExAllocatePoolWithTag(0, v14, 0x544D454Du);
    v16 = v15;
    if ( v15 )
    {
      memset(v15, 0, v14);
      sub_FFFFF80B76F32D80(v16, v14, *(_QWORD *)(v18 + 16), *(unsigned __int16 *)(v18 + 8));
      RtlInitUnicodeString(&DestinationString, v16);
      if ( RtlUnicodeStringToAnsiString(&AnsiString, &DestinationString, 1u) >= 0 )
      {
        AnsiString.Buffer = strupr(AnsiString.Buffer);
        if ( (unsigned int)IsProtectedFiles(AnsiString.Buffer)
          && v6 != 1
          && processPath
          && !strstr(processPath, "\\SYSTEM32\\CSRSS.EXE")
          && !strstr(processPath, "\\SYSTEM32\\TASKMGR.EXE") )
        {
          v5 = 0xC0000022;                      // ACCESS_DENIED
        }
```

Hidden file system

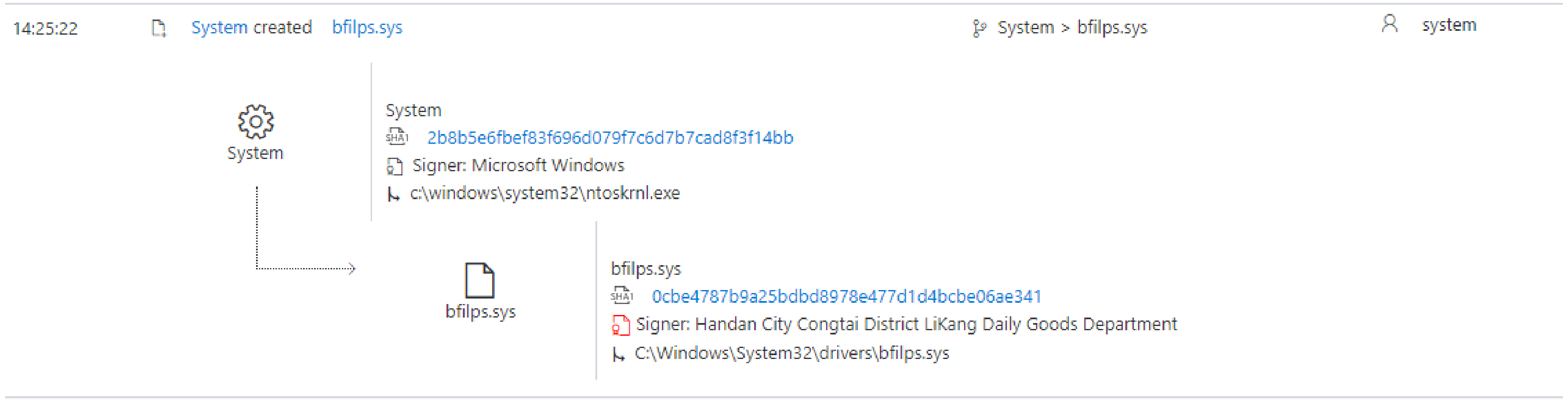
- The rootkit drivers show missing files
- When *DriverProtect* driver runs, it will:
  - Prevent access to the real file contents on the file system
  - The components reside on the hidden location have intentionally broken PE header
  - Act as a proxy and load the real contents from the protected storage after modifying the contents to be a valid PE file



✓	SiSRaid4	SiSRaid4: SiS AHCI Stor-Min...	Silicon Integrated Systems	c:\windows\system32\drivers\sisraid4.sys
✓	stexstor	stexstor: Promise SuperTrak ...	Promise Technology, Inc.	c:\windows\system32\drivers\stexstor.sys
✓	udiskMgr			File not found: system32\drivers\zcfimp.sys
✓	vsmraid	vsmraid: VIA RAID DRIVER ...	VIA Technologies Inc.,Ltd	c:\windows\system32\drivers\vsmraid.sys
✓	VSTXRAID	VIA StorX Storage RAID Con...	VIA Corporation	c:\windows\system32\drivers\vstxraid.sys
✓	WinMad	WinMad Service: Kernel Win...	Mellanox	c:\windows\system32\drivers\winmad.sys
✓	WinVerbs	WinVerbs Service: Kernel Wi...	Mellanox	c:\windows\system32\drivers\winverbs.sys
✓	xdolnkh	xdolnkh:		File not found: System32\drivers\vskudniz.sys

WDATP visibility into driver loading

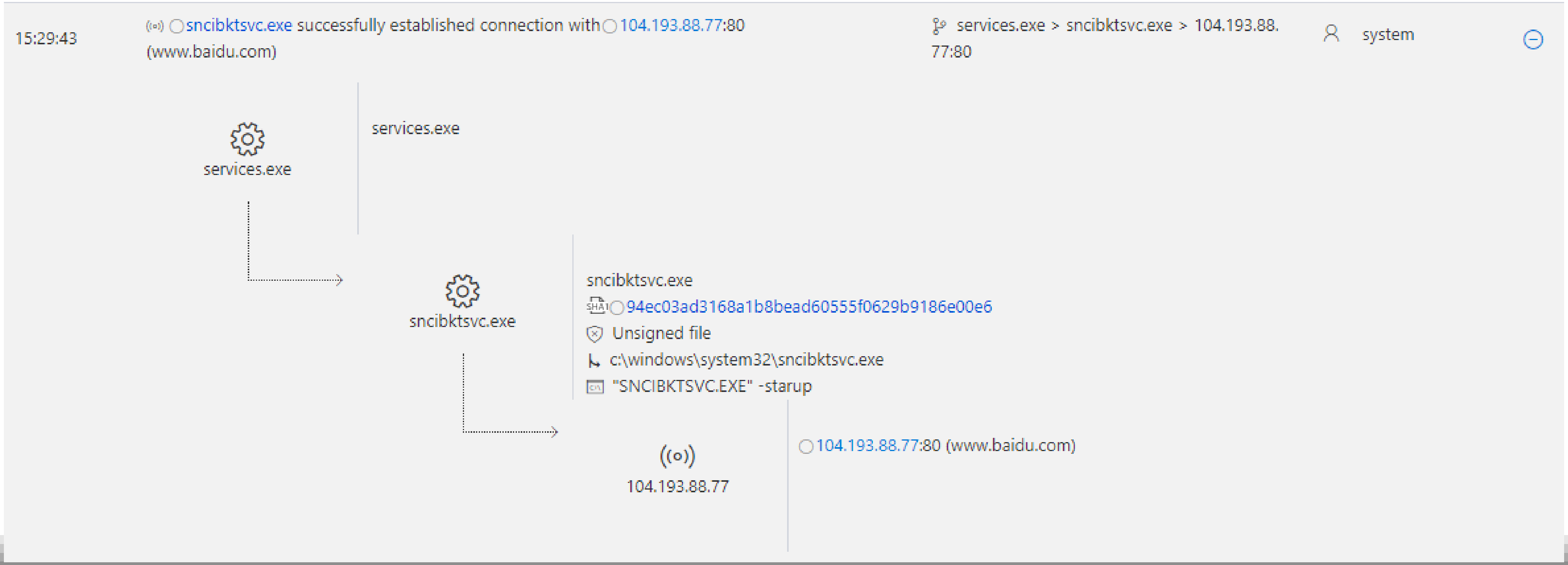
- Even though the file never touches the file system, WDATP still detects the driver loading activity
  - *DriverProtect* protection only works against user-mode tools
  - WDATP sensor works in kernel level



```
14:25:22    System created bfilps.sys                    System > bfilps.sys        system

System
  2b8b5e6fbef83f696d079f7c6d7b7cad8f3f14bb
  Signer: Microsoft Windows
  c:\windows\system32\ntoskrnl.exe

bfilps.sys
  0cbe4787b9a25bdbd8978e477d1d4bcbe06ae341
  Signer: Handan City Congtai District LiKang Daily Goods Department
  C:\Windows\System32\drivers\bfilps.sys
```

Hidden file system

- Through this Filter driver, the malware components are protected from investigations and false information on path is provided to the system. This will confuse security products and analysis tools.
- Ex) *C:\windows\system32\sncibkt.exe* image is actually *C:\Windows\System32\spsatrm\sncibkt.exe*



15:29:43 sncibktsvc.exe successfully established connection with 104.193.88.77:80 (www.baidu.com)

services.exe > sncibktsvc.exe > 104.193.88.77:80

system

services.exe

services.exe

sncibktsvc.exe

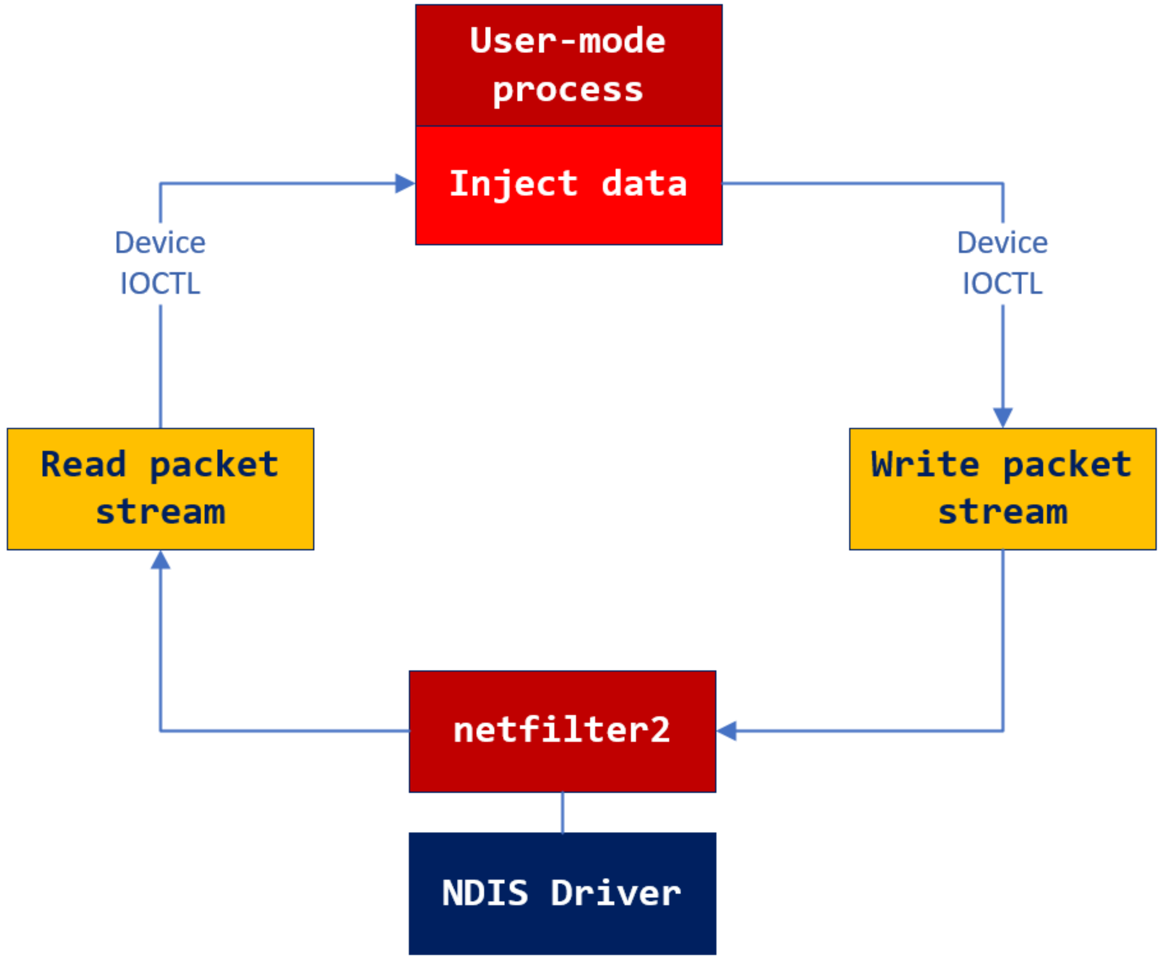
sncibktsvc.exe
94ec03ad3168a1b8bead60555f0629b9186e00e6
Unsigned file
c:\windows\system32\sncibktsvc.exe
"SNCIBKTSVC.EXE" -starup

104.193.88.77

104.193.88.77:80 (www.baidu.com)

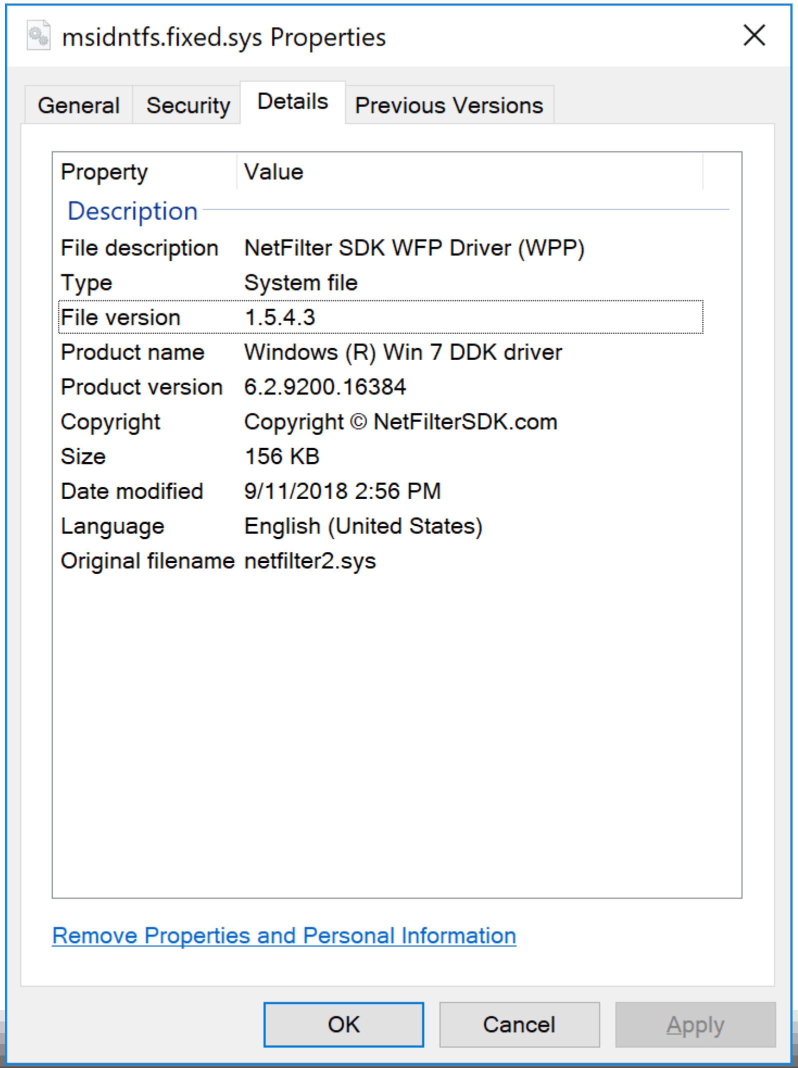
Network traffic injection

Network traffic injection



Network traffic injection – netfilter2.sys

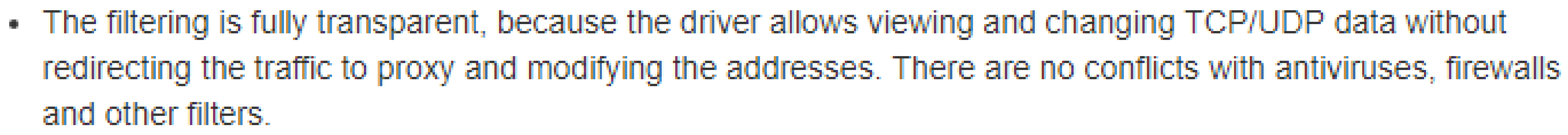
There is a MITM kernel component that are written based upon commercial netfilter2 driver code from netfiltersdk.com


msidntfs.fixed.sys Properties

Property	Value
Description	
File description	NetFilter SDK WFP Driver (WPP)
Type	System file
File version	1.5.4.3
Product name	Windows (R) Win 7 DDK driver
Product version	6.2.9200.16384
Copyright	Copyright © NetFilterSDK.com
Size	156 KB
Date modified	9/11/2018 2:56 PM
Language	English (United States)
Original filename	netfilter2.sys

Netfilter2 – transparent proxy

The filter driver provides functionality to inject packets on the fly.

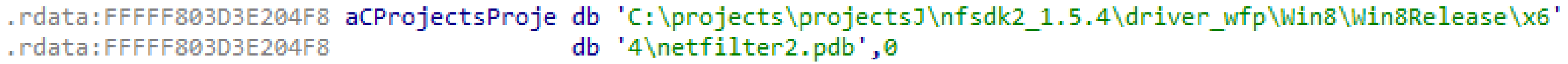


- The filtering is fully transparent, because the driver allows viewing and changing TCP/UDP data without redirecting the traffic to proxy and modifying the addresses. There are no conflicts with antiviruses, firewalls and other filters.

https://netfiltersdk.com/nfsdk.html

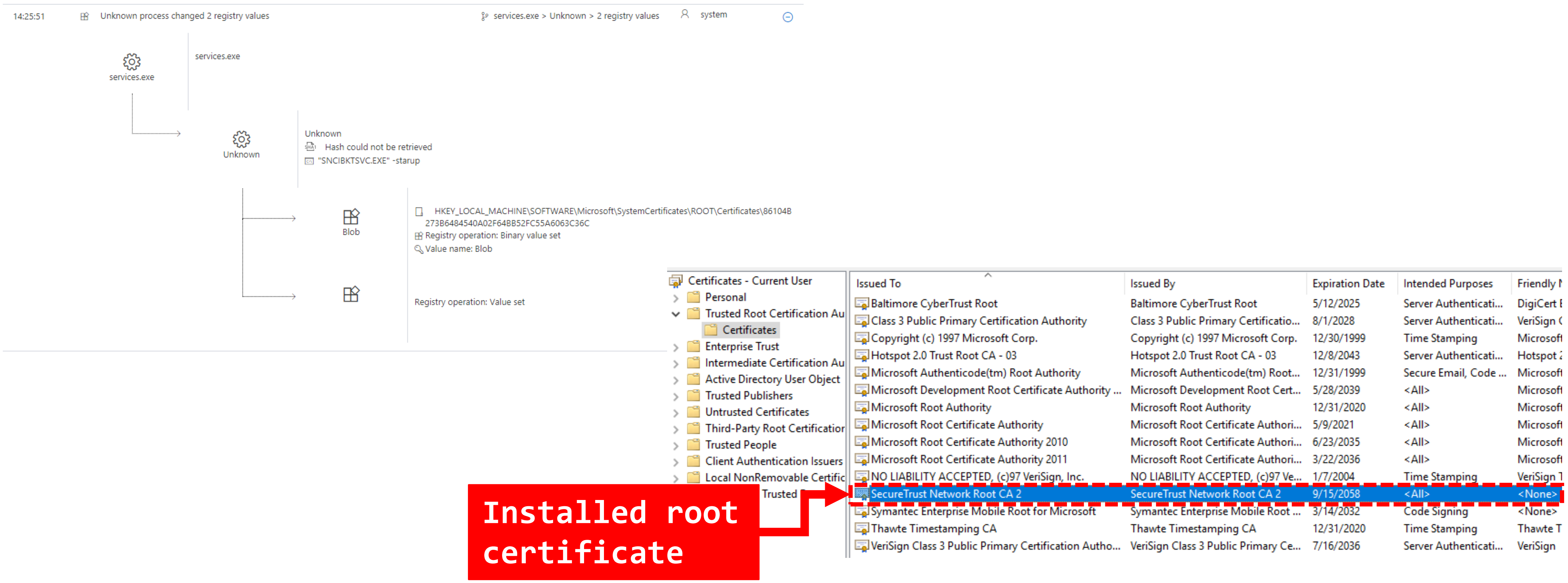
Network traffic injection – netfilter2.sys

- The netfilter2.sys driver will be loaded with random names through hidden file system
- It looks like this netfilter2.sys has close similarity to the NetFilterSDK.com provided one
- We believe the attackers have access to the netfilter2.sys source code
- The source code is commercially available



```
.rdata:FFFFF803D3E204F8 aCProjectsProje db 'C:\projects\projectsJ\nfsdk2_1.5.4\driver_wfp\Win8\Win8Release\x6'
.rdata:FFFFF803D3E204F8                 db '4\netfilter2.pdb',0
```

Adding new root certificate



14:25:51 Unknown process changed 2 registry values services.exe > Unknown > 2 registry values system

services.exe

services.exe

Unknown

Unknown
Hash could not be retrieved
"SNCIBKTSVC.EXE" -starup

Blob

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SystemCertificates\ROOT\Certificates\86104B273B6484540A02F64BB52FC55A6063C36C
Registry operation: Binary value set
Value name: Blob

Registry operation: Value set

Certificates - Current User
- Personal
- Trusted Root Certification Au
  - Certificates
- Enterprise Trust
- Intermediate Certification Au
- Active Directory User Object
- Trusted Publishers
- Untrusted Certificates
- Third-Party Root Certificatior
- Trusted People
- Client Authentication Issuers
- Local NonRemovable Certific
- Trusted

Issued To	Issued By	Expiration Date	Intended Purposes	Friendly N
Baltimore CyberTrust Root	Baltimore CyberTrust Root	5/12/2025	Server Authenticati...	DigiCert E
Class 3 Public Primary Certification Authority	Class 3 Public Primary Certificatio...	8/1/2028	Server Authenticati...	VeriSign C
Copyright (c) 1997 Microsoft Corp.	Copyright (c) 1997 Microsoft Corp.	12/30/1999	Time Stamping	Microsoft
Hotspot 2.0 Trust Root CA - 03	Hotspot 2.0 Trust Root CA - 03	12/8/2043	Server Authenticati...	Hotspot 2
Microsoft Authenticode(tm) Root Authority	Microsoft Authenticode(tm) Root...	12/31/1999	Secure Email, Code ...	Microsoft
Microsoft Development Root Certificate Authority ...	Microsoft Development Root Cert...	5/28/2039	<All>	Microsoft
Microsoft Root Authority	Microsoft Root Authority	12/31/2020	<All>	Microsoft
Microsoft Root Certificate Authority	Microsoft Root Certificate Authori...	5/9/2021	<All>	Microsoft
Microsoft Root Certificate Authority 2010	Microsoft Root Certificate Authori...	6/23/2035	<All>	Microsoft
Microsoft Root Certificate Authority 2011	Microsoft Root Certificate Authori...	3/22/2036	<All>	Microsoft
NO LIABILITY ACCEPTED, (c)97 VeriSign, Inc.	NO LIABILITY ACCEPTED, (c)97 Ve...	1/7/2004	Time Stamping	VeriSign T
SecureTrust Network Root CA 2	SecureTrust Network Root CA 2	9/15/2058	<All>	<None>
Symantec Enterprise Mobile Root for Microsoft	Symantec Enterprise Mobile Root ...	3/14/2032	Code Signing	<None>
Thawte Timestamping CA	Thawte Timestamping CA	12/31/2020	Time Stamping	Thawte T
VeriSign Class 3 Public Primary Certification Autho...	VeriSign Class 3 Public Primary Ce...	7/16/2036	Server Authenticati...	VeriSign

- The malicious user-mode component will add new root certificate
- Used to hijack HTTPS sessions on the system

Conclusion

- Detrahere (Zacinlo) is a threat that intercepts network traffic on a machine to inject ads
- It has multiple self-protection mechanisms
    - Hidden file system to hide core drivers
    - Anti-analysis/debug/detection
- It abuses feature in Windows driver verification to load kernel drivers using revoked certificate
- WDATP has a good visibility into the detailed behaviors from the threat
- WDO can be used to remediate the threat overriding persistence mechanism

C&C Servers

IP	Description
119.28.136.132:80 (gpt5.com)	ASN: 132203 City: Beijing State: Beijing Country: China Organization: TENCENT CLOUD COMPUTING (BEIJING) CO. LTD.
104.193.88.77:80 (www.baidu.com)	ASN: 55967 City: Cupertino State: California Country: United States Organization: BAIDU USA LLC
211.159.220.234:80 (adxco.cn)	ASN: 45090 City: Beijing State: Beijing Country: China Organization: TENCENT CLOUD COMPUTING (BEIJING) CO. LTD.
119.28.137.94:8080 (www.user2best.com)	ASN: 132203 City: Beijing State: Beijing Country: China Organization: TENCENT CLOUD COMPUTING (BEIJING) CO. LTD.

IOCs

SHA1	SHA256	Filename	Defender/descriptions
deb585177e3fb4a935ca177260b02714ab511353	5edeba23daabdeaaefea7d0ba3c153a8db07363a16c659cd120e3aa9981f485b	setup.exe	Infector
954e690318768729b2e825622c883b803fcb8433	bf57248c47bb1fc44bafad7bb257d1e03e04128d847e5d895a05ec83cea5bd27	C:\Windows\System32\spsatrm\sncibkt.exe	Trojan:Win64/Detrahere
94ec03ad3168a1b8bead60555f0629b9186e00e6	8f0d55b54ddccf97ea798b40fc0a921f59010e5f02118251438ffcf79f19847a	C:\windows\system32\sncibktsvc.exe	NULL filled file
86218530d9043ff51e1d581a96e89140820c8fcb	d9fcc3554d657d68c94001438ebce24842cec393ad97d3789a30c074261519ad	C:\Windows\System32\spsatrm\sncibkt.sys	VirTool:Win64/Detrahere
0cbe4787b9a25bdbd8978e477d1d4bcbe06ae341	fcbce0027b85069790b25b08444acc4ebcb24567d6f461e63ca20f067e7284e6	bfilps.sys	Trojan:Win64/Detrahere.S
1cb1f70a120a61ee9c97d8f7c5ba6e9ea8674e51	78ac863f8ccea5cd81a3361c203ba792379735ba5a311d8607f1f1e5872edb2d	lsswzcgj.sys	Trojan:Win64/Detrahere.S
9258b5d3a559ed02a4afaf0dd8079820ebff3bc8	c86de08ac277735e62bef81a3068536b43cccf8f278e6cd59e50a6a8874c4973	rtdsuxz.exe	Trojan:Win32/Detrahere.B!dr
69d209cb78d8e37de47bc697169f6bb7de4fa738	69d209cb78d8e37de47bc697169f6bb7de4fa738	notepad.exe	Trojan:Win32/Detrahere.B!dr