

TDSS.rootkit

April 11, 2012

Summary

TDSS rootkit appeared around 2008 and is known for its ability to survive in the machine without being detected and the challenges it presents in terms of cleanup. There have been four versions of TDSS before this latest variant, and there have been improvements with every version in terms of being stealthy.

Detailed information about the rootkit, propagation vector, characteristics and mitigation etc are explained in the following sections.

- Infection and Propagation Vectors
- <u>Characteristics and Symptoms</u>
- <u>Restart Mechanism</u>
- Getting Help from the McAfee Foundstone Services team

Infection and Propagation Vectors

TDSS spreads by using affiliate marketing programs. Most affiliate marketing programs spreading malicious code use a Pay Per Install model which means the amount earned by the malware author depends on the number and the location of the machines it infects.

Characteristics and Symptoms

TDSS.e!rootkit

There are multiple variants of TDSS in the wild. All these variants exhibit different behavior. These are some of the behaviors exhibited by this variant *TDSS.e!rootkit*:

Upon execution of the dropper adjusts "SE_LOAD_DRIVER_PRIVILEGE", on success copies itself as a .DLL and calls **AddPrintProcessor**, which requests the system process "SPOOLSV.EXE" to load the specified library. It then creates a random service by executing ZwLoadDriver.

The malware hooks "KiDebugRoutine" which enables the malware to hide its traces in memory from a debugging program. When a debugging program tries to access the malware traces in memory through this hook, the malware intercepts the request and points to clean memory instead of the actual malicious code.

It then infects a windows component (.SYS file) which will enable it to start during system boot. The malware injects a thread in the kernel, so whenever the infected .SYS file is requested it always returns a clean one, instead of the one infected by the malware. In addition to this, the malware redirects searches. It also connects to its command and control server and sends information and receives commands. Connections to the following domains were observed on a infected machine:

- https://nichtadden.in/
- https://91.212.226.67/
- https://li1i16b0.com/
- https://zz87jhfda88.com/
- https://n16fa53.com/
- https://01n02n4cx00.cc/
- https://lj1i16b0.com/
- http://clickpixelabn.com/
- http://thinksnotaeg.com/
- http://ijmgwarehouse.com/

- http://getbestbanner.com/
- http://pixelrotator.com/
- http://rf9akjgh716zzl.com/
- http://justgomediainc.in

TDSS.f!rootkit

Like other droppers of TDSS, TDSS.f dropper also carries actual infector in its resource section in an encrypted form. The actual infector is first decrypted and the dropper image is replaced with the decrypted infector.

Workflow Diagram for TDSS:



Ģ • 🛅 "BIN"	RC4 encrypted	
🗁 "MBR" - [lang:1033]		
🖻 🫅 "FILE"	Offset 0 1 2 3 4 5 6 7 8 9 A B C D E F Ascii	
AFFID" - [lang:1033]	00000000 B8 01 00 00 57 1A 83 A5 08 23 07 50 26 E3 CA 77 , ₩ ¥	#∎P&ãËw
🖰 "BOOT" - [lang: 1033]	00000010 A3 15 53 F1 8B EB A0 85 AC 72 47 7E 14 EF F8 F6 £∥Sñ∥ë ~	rG∼∎ïøö
CMD22" [lang:1022]	00000020 90 8D 62 7D 47 83 88 Å8 Å7 64 29 91 F4 23 B1 58 b}G `S	d)´ô#±X
	00000030 5F A7 C4 5B C3 15 BC 1B 6F 72 DF D3 27 25 2D B4 _\$A[A] ¹	rBO'%-1
"CMD64" - [lang:1033]	00000040 CD FC 0B 37 7C D5 3E 90 3F C0 41 69 65 5A A3 3F Iu 7 0> ?.	AA1eZ£?
	UUUUUU50 9D 55 18 DA 5A 08 CD F9 77 B3 A1 22 E1 CD 0B DC [e]Uj]Iuw	'I allu
	UUUUUUUUU 81 UI E5 C7 U3 ZU E8 EE 4F 75 ZU EU IU U5 83 CA aC .,%U 199999979 D9 22 21 E4 E3 90 9D 90 E7 99 72 37 D4 94 9E E1 *2147***U	u.a.L
	UUUUUU/U BU 32 21 F4 5A UC 9B 8C 56 88 72 A6 D4 94 9E F1 2!02 V 000000000 F3 1D FF D3 40 3D 70 D3 0F 34 F0 00 F1 40 04 3F 7•5682•	r;un As oger
🖰 "DRV64" - [lapg:1033]	00000000 5A ID FE D3 40 ZD /7 BZ 05 34 F0 00 5I 40 74 35 Z DOW-Y-I 00000000 74 40 4E C7 CE 3E 30 7D C0 34 70 40 4E 37 E3 E1 +10cT/01È	40.0015 5×UE7C+
	000000000 74 6C 4F C7 CF 2F 37 7D C6 26 2A 46 45 37 53 B1 01001773E 000000000 1 1 F9 10 D2 F6 32 90 00 27 9C 66 F2 P1 72 P5 F2 44000017	≪*nE/SL ∎fâtsuò
LDR32 - [lang;1033]	000000000 14 25 10 52 F0 42 50 00 57 50 00 22 51 75 55 F2 600000 7	BÉI7 ÀS
"CDR64" - [lang:1033]		Ýkå 'v'
	00000000 CB F7 AD EE 66 9C 0E 55 FF 59 62 47 7E 7C CA A3 Ê→-îfill0	VhG~lÊs
	000000E0 A8 18 F1 2C E5 6C 8D 5D FC 43 78 32 5E 14 67 6C 1	Cx2^lg1
🖻 🫅 "PAIR"	000000F0 E5 3F B2 26 A8 9C 73 56 4F 22 66 2D 03 38 63 22 a?*& svo	"f-18c"
	00000100 E6 94 3E C8 B8 05 2D E4 66 2B C4 7F CB 81 EE 87	+ÄĮĒĮî
Source [langitoco]	00000110 F2 4E E7 40 3D C4 D4 FE 10 E5 BB B9 BA 31 61 95 ÒNç@=ÄÔþ∣	å≫¹º1a∎
MAME - [Iding,1000]	00000120 B3 39 99 63 77 01 F9 8A 44 CA OF 44 A6 03 74 6D ³9∥cw∥ù∥D	Ê D tm
	00000130 2A F9 57 86 96 33 B9 7A DD A6 E4 5C 88 25 CC B9 ★ù₩II3¹zÝ	¦ä∖∎%11
Maliaious components (PC4	00000140 7C CC 0D 69 DC E3 AF B4 FC 52 28 25 2B 9C 40 FF I.iUā (ü	R(%+∎@y
Mancious components (RC4	00000150 AC 73 C2 73 66 C5 23 CF F9 45 82 40 29 3C E3 0B ¬sAsfA#Iù	E[@)<ã[
encrypted) in resource	00000160 D0 B2 BD 07 7D 01 EB 39 9B F3 0D E6 41 2E C1 3D Đ ² ½ }[e9]	Ó.æA.A=
section of the infector	000001/0 91 02 94 18 83 CA 99 D3 E4 01 3D 1F 8B 64 AE 60	
		HV KQI
	00000170 DO 4/ /D 4A 5/ 5E AF 33 AF /D 3D DE 28 20 FF 40 105JW 3 .	V-P(,y@ 1/ +∎∎∎/
	000001A0 C3 /4 31 0E FF 0A F/ 0A 41 DC 07 D1 0/ 00 90 3C Atinyj-A	74.1111

(1.1 Resource Section of infector)

These resources are loaded whenever required. Initially, it loads few resources like "BUILD", "NAME" of "PAIR" type and "SUBID", "MAIN" of type "FILE".

Some variants of TDSS.f before infection checks if it is running in controlled environment.

It connects to the "Root\Cimv2" WMI interface, retrieve system resource information like instances of Win32_BIOS, Win32_DiskDrive, Win32_SCSIController, Win32_Processor, Win32_Process and check if the malware is being executed in controlled environment.

Property of	of W	ML	nsta	nce	2								8	00	65	00	M.o.d.e.1X.e.
													5	00	16	00	n. N.a. m.e.
Model, Na	me.	and	I Ma	nuf	actu	rer							D	00	61	00	Q.E.M.UM.a.
00121704	UE		12		UU.	1111	UII		US	0.0	1.4	0.0	7.5	00	72	00	h.u.f.a.c.t.u.r.
0012F714	65	00	72	00	00	00	12	00	57	00	69	00	бE	00	33	00	e.rW.i.n.3.
0012F724	Va	hies												0	72	88	2D.i.s.k.D.r.
0012F734		nico												Ø	33	00	i.v.eW.i.n.3.
0012F744	V.	. 0	EMI	I D	d H	at I	VRO	YE	Rack	ie l	(Maa	210		6	91	7C	2B.I.O.S
0012F754	Ae	II., U	LIVIU	, n		at,	VDU	CIL	JOCI	15, V	IVI VV	are,		0	00	00	R.e.dH.a.t
0012F764	CII	пx,	VIII	uar	nv,	Cap	ture	Cire	ent.e	xe				0	6F	00	V.B.O.XB.o.
0012F774	63	00	68	00	73	00	00	00	56	00	4D	00	77	00	61	00	c.h.sV.M.w.a.
EWMI Clas	ses:	1														00	r.eW.i.n.3
0																00	2S.C.S.I.C.o.
Win32 BI	05	Wi	32	Diel	Driv		Nin ³	20 5	CSI	Con	trol	Ar				00	n.t.r.o.l.l.e.r
Min32 D	0.00		W	132	Die	LD	10	200	W	.32	Dro	CAR	-			00	C.i.t.r.i.x
Sunzin Us	ULC.	3501		1132	115	NU I	UZ.	and		00	110	111			0.1	00	V.i.r.t.u.a.
0012F7D4	60	00	20	00	48	00	44	00	00	00	Bő	00	57	00	69	00	1H.D¶ W.i.
0012F7E4	6E	00	33	00	32	88	5F	88	50	00	72	88	6F	88	63	00	n.3.2P.r.o.c.
0012F7F4	65	00	73	00	73	00	00	00	57	00	69	00	6E	00	33	00	e.s.sW.i.n.3.
0012F804	32	00	5F	00	50	00	72	00	6F	00	63	00	65	00	73	00	2P.r.o.c.e.s.
0012F814	73	00	óΕ	00	72	00	00	00	43	00	61	00	70	00	74	00	s.o.rC.a.p.t.
0012F824	75	00	72	88	65	99	43	99	60	00	69	00	65	99	óΕ	00	u.r.e.C.1.i.e.n.
0012F834	74	00	2E	00	65	88	78	00	65	00	00	00	53	00	45	00	te.x.eS.E.
0012F844	40	88	45	99	43	96	54	98	20	98	2A	00	26	98	46	90	L.E.C.T*F.
0012F854	52	00	4F	00	4D	88	20	00	25	00	73	00	20	00	57	00	R.O.M%.sW.
0012F864	48	00	45	88	52	88	45	00	20	88	25	88	73	88	20	00	H.E.R.E%.s
WMI Quer	y fo	rma	t:														=".%.5."
																	(2.S.E.L.E.C.T.
SELECT *	FRO	M %	s W	HEF	RE %	s =	%										.*F.R.O.M
SELECT *	FRO	M %	s W	HEF	RE %	sII	KE (%%s	%%								%.sW.H.E.R.E.
					1000		0000		and.								.%.sL.I.K.E.
																	.".%.%.%.5.%.%

(1.2 WMI Instance)

If the malware discovers that it is running in controlled environment,

- It skips the execution path which is responsible for MBR partition table modification and creation of hidden filesystem, wherein it keeps original MBR and other malicious components in encrypted form.
- It directly calls a routine which removes its traces by deleting the dropper and other files from the %TEMP% folder as shown below:



"\\.\PhysicalDrive0" and uses IOCTL_SCSI_PASS_THROUGH_DIRECT I/O control code to read and write to the HDD.

For data transfer operations, a buffer with alignment matching the adapter device is required.

Therefore it first retrieves AlignmentMask using the **IOCTL_STORAGE_QUERY_PROPERTY** control code and then retrieves the capacity of the device using **IOCTL_SCSI_PASS_THROUGH_DIRECT** control code.

Infection Flow

It first reads MBR into memory, parses the partition table and look for the bootable partition.

After identifying the bootable partition, it computes absolute number of sectors by adding the LBA of the first absolute sector of active partition and number of sectors in the partition.

The summation of the above two is then subtracted from 0x1000000. It then calculates the number of sectors to be used in the filesystem to be created.

Number of sectors = 0x1000000 - (LBA of first absolute sector of the active partition + Number of sectors in the partition) - <math>0x10

00	01	01	00-07	FE	FF	FF 3F	00	00	00-14	AC	FF	00	Number of sectors in
80	FE	FF	FF-17	FE	FF	FF-53	AC	FF	00-9D	53	00	00	the new partition (0x539D)
00	88	00	00-00	88	00	00-00	00	00	00-00	00	00	00	= 0x1000000 (0x3E
00	00	00	00-00	00	00	00-00	00	00	00-00	00	00	00	+ 0xFFAC1F) - 0x10
55	AA		Partit	tion	tab	le from	mod	lifie	d MBR				

(1.4 Modified MBR)

It then loads, and decrypts the resource named "**vbr**" of type "**BIN**" in memory. To avoid re-infection, it compares the malicious VBR code with the original VBR code as shown below:

edx, [esp+3F4h+var_3BC] lea Malicious VBR loaded into memory and call FindAndLoadResource RC4Decryption decrypted add esp, 4 eax, eax test loc 404000 jz edx, [esp+3F0h+var_3B8] MOV test edx, edx 10C_404D00 jz ecx, [esp+3F0h+var_3C4] ecx, 0FFFFFFACh mov add edx, 54h add add esi, 54h ecx, 4 CMD **Compare malicious and original VBR** short loc 404779 jb mov eax, [esi] ; DS:[00B62BC4]=8EC033FA <-- DWORD from the original VBR [+54] eax, [edx] ; DS:[00B64484]=8EC031FA <-- DWORD from Decrypted VBR cmp short loc_404781 jnz ecx, 4 sub edx, 4 add add esi, 4 cmp ecx, 4 short loc 404765 jnb (1.5 Code flow 2)

The original MBR which was read previously into memory is encrypted, written into the hidden file system and the storage information is saved in secondary configuration as shown below:

004011231 79 08	LINS SHORT goog eue 66481120		
86461125 46	NEC ENX	RC4 Encryption	
00401126 81CA 00FFFFFF	OR EDX FFFFF00		
3646112C b2	INC EDX		
30401120 42 30401120 0FR6F2		01=03	
80401120 8FB61086	HOUZY ERY RUTE DTD NC+FECT+FAX1		
80401130 0000000 80401134 8855 FF	MOIL RVTE PTR SS.[LSI'LEN]	DI = 03 (b) I = 07	
00401134 0033 FC 00501104	HOW DITE FIN 33.[EDF 2],DE	Stack DS-[0012F3E]=00	
BBLB1120 9910B1	MON DUTE DTD NS.[CONCHA]		
B040113H 001001	HOW DITE FIN DS.[EGATERA],DE	$D_1 = 0^2$, Stack D3. [0012[H2D] = 03	
00401130 001400 00604460 00044004	HOW DITE FIN DS.[ESITERA], DE	DL-03, SLACK DS.[0012FH2F]-09	
	HUVEN EDN,DTTE FIN VS.[EGN*EHN]	SLACK DS.[0012FH2D]-07	
00401144 0FD0D2 80664467 8959	MUVEA EDA,UL ADD EDV EDV	DL-03 DD-03	
00401147 0303	HUU EUA,EBA	EBA=00000009, EDA=00000003	
00401149 81E2 FF000080	HOU EDA, 600000FF		
	JNS SHORT GOUGLEUP. 00401159		
00401151 4H			
00401152 816H 00FFFFF			
00401158 42		DI - 00	
00401159 0FB002	HUVZA EDA,UL	DL=06	
	HOW THY NUMBER AT CONTENDED AND	Stack VS.[8012FH38]=44 (V)	
00401100 8855 08	NOV EDA, DWORD PIK 55: [EBP+8]	Stack 55:[0012F9F8]=00B04430	
00401103 301617	AUR BYIE PIR DS:[EDI+EDA],BL		
00401100 4/	ING EDI		
00401167 3B7D 0C	CMP EDI,DWURD PIR SS:[EBP+C]	STACK 22:[nalshahc]=nnannsnn	
0040116A 72 94	JB SHURI (GREENEL32.SetEndUffile)		
0040116C 8A4D FF	MOV CL,BYTE PTR SS:[EBP-1]	PC/ Encruited MBP	
BL=2E ('.')	Contents from MBR	Key Enclypeet more	
AND READ RATES AND A DATES AND	Solitonis Holli III Dit		
h2:[AAB0A048]=33 (.3.)			
DS:[00869678]=33 (3)		Address Hex dumn ASCI	
); 00809078]=33 (*3*)	ASCII ASCII AG 70 ED EG A7 EG 4E EP DE 4D 70 JANDAR JADDAD	Address Hex dump Barill Baradara 10 cs no 75 cf of ol on 55 so of crigar 57 da oc 180 útil	
PS: 00609078] = 33 (*3*) Address Hex dump 90869678 33 C0 8E D0 BC 90869678 8E 18 66 56 57	ASCII 80 7C FB 50 07 50 1F FC BE 1B 7C 3À80%, DPMP1 20 FE 01 F3 04 FB 80 8F 07 01 04 MBD4138/3420	Address Hex dump ASULI 09869678 10 C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C ■ÂÝuï@;■T]'Ě∎G@, 09860488.E2 AE 99 81 0C A8 89 E5 CB 80 23 2C 92 AE 83 DA 30■ "Ì̤# 'Aª/Ú	
305: 00609078 =33 (3*) 4ddress Hex dunp 90609078 33 C0 8E D0 BC 90609088 BF 18 66 50 57 90609088 BF 6.00 7 BC 57	ASCII 80 7C FB 50 07 50 1F FC BE 1B 7C 3À80%, IQPMP1 89 E5 01 F3 A4 CB BD BE 07 B1 04 ;umpw'âmó≈ê3 7C 13 23 C5 18 E2 E4 CD 18 29 E5 P0 L 19 20 E5	Address Hex dump ASULI 09869678 10 C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C MÅÝuï0; UT]'ĚUG0, 098696868 63 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F 83 DA ãout.''åĚU#,'o³ú 098696068 F6 F5 81 76 D5 38 9E 51 CB 8A 23 2C 92 6F 83 DA ãout.''åĚU#,'o³ú 0986960608 F6 F5 81 76 D5 38 9E 14 88 76 59 E3 13 98 úñuñúnu anuvãu	
DS: [00069678]=33 ('3') address Hex dump 80069678 33 C0 8E D0 BC 80069688 BF 18 96 50 57 80069698 38 6E 00 7C 99 80069698 38 6E 00 7C 99	ASCII 90 7C FB 50 07 50 1F FC BE 1B 7C 3ÀH0¼, IÛPMPH 89 E5 01 F3 A4 CB BD BE 07 B1 04 ; MHPW'ÂMÓ¤Ĕ3 75 13 83 C5 10 E2 F4 CD 18 8B F5 8n j.umañaô 13 93 2C 10 E6 A0 BC 07 BL 07 80 MGHT 55 1	Address Hex dump ASULI 09869678 10 C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C 1470; 117'ELGG, 09869688 63 65 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F 83 DA 3011."'AETH, '0°JÚ 09869688 65 65 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F 83 DA 3011."'AETH, '0°JÚ 09869688 65 65 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F 83 DA 3011''AETH, '0°JÚ 09869688 65 70 82 70 88 40 95 81 76 D5 30 9F 14 00 61 08 76 59 E3 13 98 úõluõuõun.auvyänn 008696968 65 70 82 80 66 47 80 55 81 76 05 30 9F 14 40 55 88 76 59 E3 13 98 úõluõuõun.auvyänn	
DS: 000000000000000000000000000000000000	ASCII 80 7C FB 50 07 50 1F FC BE 1B 7C 3ÀH94.10PMP1 89 E5 01 F3 A4 CB BD BE 07 B1 04 ;MHPW'âH64Ë 75 13 83 C5 10 E2 F4 CD 18 8B F5 8n j.uHMÅHå 19 38 2C 74 F6 A0 B5 07 B4 07 88 MÆHILB,tö j 5 PD 07 A6 PL 05 CD 14 5P 69 80 50 ft.	Address Hex dunp ASCII 09869678 1D C5 DD 75 CF AE AI 9D 54 5D 92 CB 08 47 40 2C MÅVIÖS; HT]'ÉHGG, 09869678 1D C5 DD 75 CF AE AI 9D 54 5D 92 CB 08 47 40 2C MÅVIÖS; HT]'ÉHGG, 09869688 E3 6F 99 81 9C A8 89 E5 CB 8A 23 2C 92 6F B3 DA 30 ML ·>30 10 ·>30 96 70 MUÉH, 'o 3'Ú ·>30 10 ·>30 98 10 ·>30 98 10 10 ·>30 10 ·>30 98 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10<	
DS: L0066 90 78 33 (*3*) 1ddress Hex dunp 90869678 33 C8 E D8 90869678 33 C8 E D8 D5 90869678 33 C6 92 D5 D5 90869688 BF 18 96 D6 D7 99 90869688 F6 AC C6 19 97 4 90869688 F6 AC C6 19 97 4 90869688 F6 AC C6 19 4 4 90869698 F6 AC C6 19 4 4 90869698 F6 AC C6 9 6 6	ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3ÀHĐ¼, ÔPHPH 89 E5 01 F3 A4 CB BD BE 07 B1 04 ; HHPH'ÂHĞKÊ 75 13 03 C5 10 E2 F4 CD 10 80 F5 8n.].UHHÂHÂĆ 19 38 2C 74 F6 A0 B5 07 84 07 88 HÆHItB,tö FC BB 07 00 84 08 CD 10 EB F2 88 ŏ<.ti>	Image: Second state Address Hex dump ASCII U3401 00869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 0B 47 40 2C AŠVII©;IT]'ĒGQ, 00869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 0B 47 40 2C AŠVII©;IT]'ĒGQ, 00869688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F B3 DA ãout'šĒIH,'o ³ Ú G0869698 FA F5 81 76 D5 30 9F 14 08 61 08 76 59 E3 13 98 úñluõinu.auvšinu 00869688 FA F5 81 76 D5 30 9F 14 08 61 08 76 59 E3 13 98 úñluõinu.auvšinu Wuềnúclušiu(úbřu 00869688 S7 98 C8 8A FA 47 8F 44 5E BE DC 3C DA D8 E8 79 74 8% mở čá-bušiu(úbřu Wuềnúclušiu(úbřu 0086988 S0 1 20 6D F0 B4 26 F0 AC DE 1C E2 CC 86 97 74 8% mở čá-bušiu(úbřu) Wuềnúclušiu(úbřu)	
DS: 0006 9078 33 (*3*) 1ddress Hex dunp 90869678 33 C0 8E D0 90869678 33 C0 8E D0 BC 90869678 33 C0 8E D0 BC 90869678 38 GE 90 TC 99 90869698 BF 18 66 58 57 908696968 FO AC 30 49 74 90869968 FO AC 30 74 406 90 90869968 98 FO 86 76 46 74 90869968 98 76 46 60 74 90869968 98 76 46 60 74 90869968 98 76 46 60 74	ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3ÀBD¼. 0PMPH B9 E5 01 F3 A4 CB BD BE 07 B1 04 ;MMPM'âMó×ʽ 75 13 83 C5 10 E2 F4 CD 18 8B F5 8n. .uMMâMâĉ 19 38 2C 74 F6 A0 B5 07 B4 07 8B MÆILtB,Cŏ J FC BB 07 00 B4 0E CD 10 EB F2 88 ŏ≺.tŭ≫M.'MI 73 2A FE 46 10 80 7E 04 0B 74 0B NMÈF.s*þFMM'	Image: Second state Address Hex dump ASCII UX401 00869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 0B 47 40 2C #Å ⁴ uï®; #T]'ÉEG@, Mm21 90869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 0B 47 40 2C #Å ⁴ uï®; #T]'ÉEG@, Mm21 90869688 E3 6F 99 81 9C A8 B9 E5 CB 8A 23 2C 92 6F B3 DA	
DS: 000609078 33 (*3*) 1ddress Hex dunp 80869678 33 C0 8E D0 80869678 33 C0 8E D0 BC 80869678 33 C0 8E D0 BC 80869678 38 GE 90 TC 99 80869678 83 C6 19 49 74 80869668 F0 AC 3C 69 74 80869678 83 C6 19 49 74 80869678 86 F0 AC 30 74 80869678 86 F0 AC 30 74 80869678 80 7E 84 68 89 80869678 80 7E 84 80 74 80869678 80 7E 84 80 74	ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3ÀHĐ¼.]ÔPUPU B9 E5 01 F3 A4 CB BD BE 07 B1 04 75 13 83 C5 10 E2 F4 CD 18 8B F5 Sn.].uHUÃHÂC 19 38 2C 74 F6 A0 B5 07 B4 07 8B MÆLITB.C0 J 73 2A FE 46 10 80 7E B4 08 74 08 73 2A FE 46 10 80 7E 84 08 74 08 95 A0 86 07 75 280 46 02 26 83 "M.t. ¶UDU"	Image Image <th< th=""></th<>	
DS: 000609078 33 (*3*) 1ddress Hex dunp 80869678 33 CØ 8E DØ 80869678 33 CØ 8E DØ BC 80869678 33 CØ 8E DØ BC 80869678 33 CØ 8E DØ T 808696988 BF 18 06 50 T 80869688 FØ AC CØ 90 10869668 FØ AC CØ 74 80869608 FØ AC CØ FØ AG C 74 80869608 FØ AG FØ AG C 74 80869608 FØ AG BG AG SG 74 80869668 AG BG BG BG SG 57 80869668 BG BG BG BG BG SG 74 80869668 BG	ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3ÀHĐ¾, ÔPUPU B9 E5 01 F3 A4 CB BD BE 07 B1 04, ¿MUPU ⁴ àmó×Ё½ 75 13 83 C5 10 E2 F4 CD 18 8B F5 Sn.j.umuñáàá 19 38 2C 74 F6 A0 B5 07 B4 07 8B 76 BB 07 00 B4 0E CD 10 EB F2 88 5≺.tü≫M.´mí 73 2A FE 46 10 80 7E 04 08 74 08 NWÈr.s*pFMU [*] 05 A0 86 07 75 02 89 46 02 06 83 W ^{**} .tu ¶uU.àť.i 06 08 E8 21 00 73 05 A0 86 07 E0 40 BC 70 80 40	Image: Second state Hex dump ASCII 08869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C ASCII 09869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C AŠVI©; IT]'ËIGQ, 09869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C AŠVI©; IT]'ËIGQ, 09869688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F B3 DA ãout'šĒU#, 'o³Ú 09869698 FA F5 81 76 D5 30 9F 14 00 61 08 76 59 E3 13 9B úõtuõ0000auvYãnu 09869688 S3 D1 20 6D F0 B4 26 F0 AC DE 1C E2 CC 86 97 74 8Ñ mð &* abaltut MuÈuúcUJ'¾UúØēy 08869668 38 D1 20 6D F0 B4 26 F0 AC DE 1C E2 CC 86 97 74 8Ñ mð &* abaltut 98669608 46 B9 33 43 D9 B4 A6 98 CD 80 8E 9F F3 4B 98 1A F '3CD' ¦ IÍIIIIÓKINI 09869668 CD F8 5A 89 74 0D A1 4E 64 E3 E4 CD CF 68 77 CF 60 HMINDZ(.ûtÙrh'ü' 09869668 H0 B1 F1 15 D0 5A 28 00 F8 8E D9 72 68 27 FC 60 HMINDZ(.ûtÙrh'ü' 0 H0 AB 1F 15 D0 5A 28 00 F8 8E D9 72 68 27 FC 60 HMINDZ(.ûtÙrh'ü'	
DS: 000604078 33 (*3*) 1ddress Hex dump 80869678 33 C0 8E D0 BC 80869678 33 C0 8E D0 BC 80869678 33 C0 8E D0 BC 80869678 35 C6 10 49 74 80869688 F0 AC 30 74 30869688 F0 AC 74 80869688 F0 AC 30 AC 74 308696968 F0 AC 74 808696988 F0 AC 30 F0 74 808696988 F0 AC 30 F0 74 808696988 F0 AC C0 F0 74 808696988 F0 AC F0 F0 74 808696988 F0 AC F0 F0 74 808696988 F0 F0 F0 <th>ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3ÅH0½.]ÔPUPU B9 E5 01 F3 A4 CB BD BE 07 B1 04, ;umpuråmóײ½ 75 13 83 C5 10 E2 F4 CD 18 8B F5 8n.].umuñáčá 19 38 2C 74 F6 A0 B5 07 B4 07 8B FC BB 07 00 B4 0E CD 10 EB F2 88 5-<.tu> 5- 73 2A FE 46 18 80 7E 84 08 74 08 NuềF.s*bFuu 10 18 86 77 88 05 A0 86 07 75 D2 89 46 02 06 83 111.4 05 A0 86 07 74 08 80 7E 18 08 64 07 65 88 111.4 10 A0 87 20 75 02 89 46 02 76 83 40 173 2A FE 46 18 80 7E 88 46 02 66 83 111.4 185 A0 86 07 75 D2 89 46 02 76 88 30 111.4 19 00 86 82 10 87 305 A8 86 07 78 65 19 00 80 74 08 80 75 10 80 66 77 68 120 78.5</th> <th>Haddress Hex dump ASCII UX4 08869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C ASCII 09869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C Afviï8; IT]'ÊIGQ, 09869688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F 83 DA Aout "18ÊIH, '0 3Ú 09869688 FA F5 81 76 D5 30 9F 14 00 61 08 76 59 E3 13 9B Wiñuữ0000.auvYãN 09869688 S3 D1 20 6D F0 B4 26 F0 AC DE 1C E2 CC 86 97 74 8Ñ mð & **********************************</th>	ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3ÅH0½.]ÔPUPU B9 E5 01 F3 A4 CB BD BE 07 B1 04, ;umpuråmóײ½ 75 13 83 C5 10 E2 F4 CD 18 8B F5 8n.].umuñáčá 19 38 2C 74 F6 A0 B5 07 B4 07 8B FC BB 07 00 B4 0E CD 10 EB F2 88 5-<.tu> 5- 73 2A FE 46 18 80 7E 84 08 74 08 NuềF.s*bFuu 10 18 86 77 88 05 A0 86 07 75 D2 89 46 02 06 83 111.4 05 A0 86 07 74 08 80 7E 18 08 64 07 65 88 111.4 10 A0 87 20 75 02 89 46 02 76 83 40 173 2A FE 46 18 80 7E 88 46 02 66 83 111.4 185 A0 86 07 75 D2 89 46 02 76 88 30 111.4 19 00 86 82 10 87 305 A8 86 07 78 65 19 00 80 74 08 80 75 10 80 66 77 68 120 78.5	Haddress Hex dump ASCII UX4 08869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C ASCII 09869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C Afviï8; IT]'ÊIGQ, 09869688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F 83 DA Aout "18ÊIH, '0 3Ú 09869688 FA F5 81 76 D5 30 9F 14 00 61 08 76 59 E3 13 9B Wiñuữ0000.auvYãN 09869688 S3 D1 20 6D F0 B4 26 F0 AC DE 1C E2 CC 86 97 74 8Ñ mð & **********************************	
N: 0000 + 0078 33 (*3*) 1ddress Hex dunp 80869678 33 C0 8E D0 BC 80869678 33 C0 8E D0 BC 80869678 83 C0 8E D0 BC 80869678 83 C6 19 47 80869688 F0 AC C0 97 80869688 F0 AC C0 74 80869608 F0 AC C0 74 80869608 S0 TE 84 60 80869608 S0 TE 40 C 80869608 S0 TE 74 80869668 80869668 S0 S0 S0 56 80869708 </th <th>90 7C FB 50 07 50 1F FC BE 1B 7C 3Àm9%, ÔPUPH B9 E5 61 F3 A4 CB BD BE 07 B1 04 ; UUPV¹ àmó×iž F5 13 83 C5 10 E2 F4 CD 18 8B F5 81 04 ; UUPV¹ àmó×iž 19 38 2C 74 F6 A0 B5 07 B4 07 8B UFMITTB 16 17 33 35 -1 0.00 18 8B F5 86 1.1 1.00 36 18 16 18 88 7 13 10 1.00 13 16 16 16 16 16 16 16 18 16 16 18 16 16 16 16 16 16 18 16 16 16 16 16 16</th> <th>Image: Second state Hex dump ASCII UX4 00869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C ASCII 00869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C AŠVI®; IT]'ËRG@, 00869688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F B3 DA ãout'ŝËU#, '0³Ú 1 00869688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F B3 DA ãout'ŝËU#, '0³Ú 1 09869688 E3 6F 99 81 76 D5 30 9F 14 00 61 08 76 59 E3 13 9B úõtuõ0000auvyãU 1 09869688 S7 9B C8 8A FA 47 8F 4A 5E BE DC 3C DA D8 EB 79 MuÈuúGJ/¾üćÚBėy 09869688 S8 D1 20 6D F0 B4 26 F0 AC DE 1C E2 CC 86 97 74 8Ñ mð'&ð-ÞMÌÌIHT 90869688 CF B 5A 89 74 0D A1 4E 64 E3 E4 CD CF 68 77 CF ÍúZIL ; MAÏÌIHVÍ 09869688 CF B 5A 89 74 0D A1 4E 64 E3 E4 CD CF 68 77 CF ÍúZIL ; MAÏÌIHVÍ HMIHÐZ(.ûUÙrh'Ü' 1 09869688 CF B 5A 89 74 0D A1 4E 64 E3 E4 CD CF 68 77 CF 60 HMIHÐZ(.ûUÙrh'Ü' HMIHÐZ(.ûUÙrh'Ü' 1 09869688 B BF 79 98 F6 DE 31 6 C1 CB B1 BF 80 2A C2 37 G ÍúEIII ; MAĨÌIIVÍ 1 098696788 AF 77 76 61 9 AE AD 89 66 A4 F8 55 E9 78 2A D4 20 G ¶ÉHONÍAÍɱ; °AÂ7 1 098696788 AF 77 76 61 9 AE AD 89 66 A4 F8 55 E9</th>	90 7C FB 50 07 50 1F FC BE 1B 7C 3Àm9%, ÔPUPH B9 E5 61 F3 A4 CB BD BE 07 B1 04 ; UUPV ¹ àmó×iž F5 13 83 C5 10 E2 F4 CD 18 8B F5 81 04 ; UUPV ¹ àmó×iž 19 38 2C 74 F6 A0 B5 07 B4 07 8B UFMITTB 16 17 33 35 -1 0.00 18 8B F5 86 1.1 1.00 36 18 16 18 88 7 13 10 1.00 13 16 16 16 16 16 16 16 18 16 16 18 16 16 16 16 16 16 18 16 16 16 16 16 16	Image: Second state Hex dump ASCII UX4 00869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C ASCII 00869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C AŠVI®; IT]'ËRG@, 00869688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F B3 DA ãout'ŝËU#, '0 ³ Ú 1 00869688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F B3 DA ãout'ŝËU#, '0 ³ Ú 1 09869688 E3 6F 99 81 76 D5 30 9F 14 00 61 08 76 59 E3 13 9B úõtuõ0000auvyãU 1 09869688 S7 9B C8 8A FA 47 8F 4A 5E BE DC 3C DA D8 EB 79 MuÈuúGJ/¾üćÚBėy 09869688 S8 D1 20 6D F0 B4 26 F0 AC DE 1C E2 CC 86 97 74 8Ñ mð'&ð-ÞMÌÌIHT 90869688 CF B 5A 89 74 0D A1 4E 64 E3 E4 CD CF 68 77 CF ÍúZIL ; MAÏÌIHVÍ 09869688 CF B 5A 89 74 0D A1 4E 64 E3 E4 CD CF 68 77 CF ÍúZIL ; MAÏÌIHVÍ HMIHÐZ(.ûUÙrh'Ü' 1 09869688 CF B 5A 89 74 0D A1 4E 64 E3 E4 CD CF 68 77 CF 60 HMIHÐZ(.ûUÙrh'Ü' HMIHÐZ(.ûUÙrh'Ü' 1 09869688 B BF 79 98 F6 DE 31 6 C1 CB B1 BF 80 2A C2 37 G ÍúEIII ; MAĨÌIIVÍ 1 098696788 AF 77 76 61 9 AE AD 89 66 A4 F8 55 E9 78 2A D4 20 G ¶ÉHONÍAÍɱ; °AÂ7 1 098696788 AF 77 76 61 9 AE AD 89 66 A4 F8 55 E9	
N: 0000 + 9078 33 (*3*) 1ddress Hex dunp 80869678 33 C0 8E D0 BC 80869678 33 C0 8E D0 BC 80869678 87 C0 8E D0 BC 80869678 83 C6 19 47 80869688 F0 AC 3C 08 74 80869688 F0 AC 3C 08 74 80869608 F0 AC 3C 08 74 80869608 S0 TE 10 E8 46 88 80869608 S0 TE 44 67 74 80869668 C6 13 E F 70 80869668 C6 13 E F 70 80869668 C7 13 E F 70 80869678 C7 78 49 88 <th>90 7C FB 50 07 50 1F FC BE 1B 7C 3Àm0%, jûPuPu B9 E5 61 F3 A4 CB BD BE 07 B1 04 ;umPu*âmó¤ž× 75 13 83 C5 10 E2 F4 CD 18 8B F5 8n.j.uumåmåč 19 38 2C 74 F6 A0 B5 07 B4 07 8B uf#itm8,tö j FC BB 67 00 B4 0E CD 18 BF 5 - .timuu .uumåmåč 73 2A FE 46 18 80 7E 04 9B 74 08 Nie F.s.s*pFmu" 73 2A FE 46 18 87 75 28 64 62 66 83 u".tim".timuu .im".timuu .im".timuu .im".timuu .imuu</th> <th>Image: Second state Hex dump ASCII Image: Second state Address Hex dump ASCII Image: Second state OBB69678 ID C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C ASCII Image: Second state OBB69678 ID C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C ASCII Image: Second state OBB69688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F B3 DA 30 0F Image: Second state Im</th>	90 7C FB 50 07 50 1F FC BE 1B 7C 3Àm0%, jûPuPu B9 E5 61 F3 A4 CB BD BE 07 B1 04 ;umPu*âmó¤ž× 75 13 83 C5 10 E2 F4 CD 18 8B F5 8n.j.uumåmåč 19 38 2C 74 F6 A0 B5 07 B4 07 8B uf#itm8,tö j FC BB 67 00 B4 0E CD 18 BF 5 - .timuu .uumåmåč 73 2A FE 46 18 80 7E 04 9B 74 08 Nie F.s.s*pFmu" 73 2A FE 46 18 87 75 28 64 62 66 83 u".tim".timuu .im".timuu .im".timuu .im".timuu .imuu	Image: Second state Hex dump ASCII Image: Second state Address Hex dump ASCII Image: Second state OBB69678 ID C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C ASCII Image: Second state OBB69678 ID C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C ASCII Image: Second state OBB69688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F B3 DA 30 0F Image: Second state Im	
DS: L0066 90 78 J=33 (*3*) 1ddress Hex dunp 90869678 33 C0 8E D0 90869678 33 C0 8E D0 BC 90869678 33 C0 8E D0 BC 90869678 35 C0 97 P0 9	90 7C FB 50 07 50 1F FC BE 1B 7C 3ÀĐ0¼, 1ÔPHPI 89 F5 01 F3 A4 CB DD BE 07 B1 84 ;#HPM*3#6*E?; 75 13 83 C5 16 E2 F4 CD 18 85 87 81 88 <	Image: Second state Hex dump ASCII Image: Second state Address Hex dump ASCII Image: Second state OBB69678 ID C5 DD 75 CF AE AI 90 54 50 92 CB 0B 47 40 2C MÅvI@; IT]'EIG@, 301 Image: Second state OBB69688 E3 6F 99 81 0C A8 B9 E5 CB 8A 23 2C 92 6F B3 DA 30 II'' EIG@, 301 300 II'' EIG@, 300 300 II'' EIG@, 301 300 II'' EIG@, 301 300 II'' EIG@, 301 300	
No. No. <th>ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3ÀB0¼.10PMP1 B9 E5 01 F3 A4 CB BD BE 07 B1 04 ;HMPM'âMôxÊ 75 13 83 C5 10 E2 F4 CD 18 88 F5 8n.1.uUNÂMâC 19 38 2C 74 F6 A0 B5 07 84 07 8B HAUIU8,cö FC BB 07 00 B4 08 CD 10 EB F2 88 o< 73 2A FE 46 10 80 7E 04 08 74 08 95 A0 66 07 75 D2 80 46 02 C6 83 u".tu ¶uù 95 A0 86 21 00 73 05 A0 86 07 EB FUU FU 55 AA 74 08 80 7E 18 08 74 C8 A0 85 FU 72 23 8A C1 24 3F 98 8A DE 8A FC 60 0 61 06 02 EE 42 F7 E2 39 56 72 23 8A C1 24 3F 98 8A DE 8A FC 86 06 73 1C 88 01 22 E4 2 F7 E2 39 56 72 23 8A C1 24 3F 98 8A DE 8A FC 86 08 73 1C 68 01 62 28 80 97.0 86 09 73 12 68 71 162 00 68 70.0 86 09 73 12 68 71 162 00 68 80 70.0 87 16 68 70.0 80 97.0</th> <th>Image: Second state Address Hex dump Asstitution UX41 99869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 0B 47 40 2C Mávü®; IT]'ÉIG@, UX41 99869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 0B 47 40 2C Mávü®; IT]'ÉIG@, UX41 99869688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F B3 DA 3000000000000000000000000000000000000</th>	ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3ÀB0¼.10PMP1 B9 E5 01 F3 A4 CB BD BE 07 B1 04 ;HMPM'âMôxÊ 75 13 83 C5 10 E2 F4 CD 18 88 F5 8n.1.uUNÂMâC 19 38 2C 74 F6 A0 B5 07 84 07 8B HAUIU8,cö FC BB 07 00 B4 08 CD 10 EB F2 88 o< 73 2A FE 46 10 80 7E 04 08 74 08 95 A0 66 07 75 D2 80 46 02 C6 83 u".tu ¶uù 95 A0 86 21 00 73 05 A0 86 07 EB FUU FU 55 AA 74 08 80 7E 18 08 74 C8 A0 85 FU 72 23 8A C1 24 3F 98 8A DE 8A FC 60 0 61 06 02 EE 42 F7 E2 39 56 72 23 8A C1 24 3F 98 8A DE 8A FC 86 06 73 1C 88 01 22 E4 2 F7 E2 39 56 72 23 8A C1 24 3F 98 8A DE 8A FC 86 08 73 1C 68 01 62 28 80 97.0 86 09 73 12 68 71 162 00 68 70.0 86 09 73 12 68 71 162 00 68 80 70.0 87 16 68 70.0 80 97.0	Image: Second state Address Hex dump Asstitution UX41 99869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 0B 47 40 2C Mávü®; IT]'ÉIG@, UX41 99869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 0B 47 40 2C Mávü®; IT]'ÉIG@, UX41 99869688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F B3 DA 3000000000000000000000000000000000000	
No. No. <th>ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3ÀĐ\$¼, ÔPHPH B9 E5 01 F3 A4 CB DD BE 07 B1 B4 ; MHPM' ÂHĞ&Ê; 75 13 83 C5 16 E2 F4 CD 18 8E F5 8n.].uHPM' ÂHĞ&Ê; 19 38 2C 74 F6 A0 85 07 84 07 8B METIT8, co 1 73 2A FE 46 18 89 7E 84 98 8 5 - 4B 64 98 64 92 68 5 - Muñt 5 - 4B 64 98 97 40 88 M"LIT8, co 1 1 7 5 5 6 6 7 5 5 - 4 60 6 6</th> <th>Image: Second state Address Hex dump Asstitution III 99869678 1D C5 DD 75 CF AE A 90 A 40 2C MAYUIS; III III IIII 99869678 1D C5 DD 75 CF AE A 90 C B A 40 2C MAYUIS; IIII IIIII IIIII 998696968 E3 6F 99 81 90 C B A A 2C 92 6F B3 DA IIIII IIIIII IIIIIII IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII</th>	ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3ÀĐ\$¼, ÔPHPH B9 E5 01 F3 A4 CB DD BE 07 B1 B4 ; MHPM' ÂHĞ&Ê; 75 13 83 C5 16 E2 F4 CD 18 8E F5 8n.].uHPM' ÂHĞ&Ê; 19 38 2C 74 F6 A0 85 07 84 07 8B METIT8, co 1 73 2A FE 46 18 89 7E 84 98 8 5 - 4B 64 98 64 92 68 5 - Muñt 5 - 4B 64 98 97 40 88 M"LIT8, co 1 1 7 5 5 6 6 7 5 5 - 4 60 6 6	Image: Second state Address Hex dump Asstitution III 99869678 1D C5 DD 75 CF AE A 90 A 40 2C MAYUIS; III III IIII 99869678 1D C5 DD 75 CF AE A 90 C B A 40 2C MAYUIS; IIII IIIII IIIII 998696968 E3 6F 99 81 90 C B A A 2C 92 6F B3 DA IIIII IIIIII IIIIIII IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	
No. No. <th>00 7C FB 50 07 50 1F FC BE 1B 7C 3Àmb¼, ÔPMPH B9 E5 01 F3 A4 CB DD BE 07 B B4 ¿MMPM'âmó×É2 75 13 83 C5 16 E2 F4 CD 18 8B F5 8n.].ummåmåč 19 38 2C 74 F6 A0 B5 07 B4 07 8B Matitus, co j 73 2A FE 46 19 89 7E 84 08 74 68 08 7 1B 73 2A FE 46 19 89 7E 84 08 74 08 08 7 1M 95 A9 66 67 75 28 46 86 67 68 74 14 90 14 90 1M 14<th>Image: Solution of the state of th</th></th>	00 7C FB 50 07 50 1F FC BE 1B 7C 3Àmb¼, ÔPMPH B9 E5 01 F3 A4 CB DD BE 07 B B4 ¿MMPM'âmó×É2 75 13 83 C5 16 E2 F4 CD 18 8B F5 8n.].ummåmåč 19 38 2C 74 F6 A0 B5 07 B4 07 8B Matitus, co j 73 2A FE 46 19 89 7E 84 08 74 68 08 7 1B 73 2A FE 46 19 89 7E 84 08 74 08 08 7 1M 95 A9 66 67 75 28 46 86 67 68 74 14 90 14 90 1M 14 <th>Image: Solution of the state of th</th>	Image: Solution of the state of th	
No. No. <th>00 7C FB 50 07 50 1F FC BE 1B 7C 3Àmb¼, ÔPHPH B9 E5 01 F3 A4 CB DD BE 07 B D4 JUPM*ÂmôxÉ2 75 13 83 C5 16 E2 F4 CD 18 8B F5 8n.].umm%âmâč 19 38 2C 74 F6 A0 B5 07 B4 07 8B Matths, io ju 73 2A FE 46 18 80 7E 84 08 74 08 Nmatths, io ju 73 2A FE 46 18 87 E 84 08 74 08 \mbth{matths, io ju im N F5 Sa N Mbths, sa im im N F5 Sa N Mbths, sa Sa Sa Sa Sa Sa</th> <th>Image Address Hex dump Address Hex dump Address Hex dump UX41 09869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C A \$\frac{2}{3} U\$ \$\frac{1}{1} U\$ \$\frac{1}{1} U\$ <</th>	00 7C FB 50 07 50 1F FC BE 1B 7C 3Àmb¼, ÔPHPH B9 E5 01 F3 A4 CB DD BE 07 B D4 JUPM*ÂmôxÉ2 75 13 83 C5 16 E2 F4 CD 18 8B F5 8n.].umm%âmâč 19 38 2C 74 F6 A0 B5 07 B4 07 8B Matths, io ju 73 2A FE 46 18 80 7E 84 08 74 08 Nmatths, io ju 73 2A FE 46 18 87 E 84 08 74 08 \mbth{matths, io ju im N F5 Sa N Mbths, sa im im N F5 Sa N Mbths, sa Sa Sa Sa Sa Sa	Image Address Hex dump Address Hex dump Address Hex dump UX41 09869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C A \$\frac{2}{3} U\$ \$\frac{1}{1} U\$ \$\frac{1}{1} U\$ <	
No. No. <th>00 7C FB 50 07 50 1F FC BE 1B 7C 3Àmb¼, jûPuPu B9 E5 01 F3 A4 CB DD BE 07 B1 04 JUPUN ÂmôxÉ2 75 13 83 C5 16 E2 F4 CD 18 8B F5 \$n.j.umuMãdô JUPUN ÂmôxÉ2 19 38 2C 74 F6 A0 B5 07 B4 07 8B MELITB, Cô J 73 2A FE 46 18 80 7E 84 08 74 08 MELITB, Cô J 73 2A FE 46 18 07 08 46 02 68 3 "M.t< ¶U MEF.s*bFMU" 95 A0 86 07 75 2 80 46 80 74 08 MET.t ¶U MEF.s*bFMU" \$U \$U</th> <th>Image: Solution Address Hex dump Address Hex dump Address Hex dump Address Address Address Hex dump Address Addres Address Address</th>	00 7C FB 50 07 50 1F FC BE 1B 7C 3Àmb¼, jûPuPu B9 E5 01 F3 A4 CB DD BE 07 B1 04 JUPUN ÂmôxÉ2 75 13 83 C5 16 E2 F4 CD 18 8B F5 \$n.j.umuMãdô JUPUN ÂmôxÉ2 19 38 2C 74 F6 A0 B5 07 B4 07 8B MELITB, Cô J 73 2A FE 46 18 80 7E 84 08 74 08 MELITB, Cô J 73 2A FE 46 18 07 08 46 02 68 3 "M.t< ¶U MEF.s*bFMU" 95 A0 86 07 75 2 80 46 80 74 08 MET.t ¶U MEF.s*bFMU" \$U \$U	Image: Solution Address Hex dump Address Hex dump Address Hex dump Address Address Address Hex dump Address Addres Address Address	
No. No. <th>80 7C FB 50 07 50 1F FC BE 1B 7C 3Àm0%, jûPuPu B9 E5 61 F3 A4 CB DD BE 07 B1 04 Jumpu/Juó×É2 75 13 83 C5 10 E2 F4 CD 18 8B F5 Sn.j.umuMädá 19 38 2C 74 F6 A0 B5 07 B4 07 8B mElitBs, Cö Jumpu/Jaúsé2 73 2A FE 46 18 80 7E 84 08 74 08 NuÈF.s=bFUT MI 73 2A FE 46 18 80 7E 84 08 74 08 NuÈF.s=bFUT MI NuÈF.s=bFUT MI NuÈF.s=bFUT 9 08 66 63 M".tt<<td>41 41 9 18 18 14 18 14 08 14</td></th> <th>Image: Solution of the state of th</th>	80 7C FB 50 07 50 1F FC BE 1B 7C 3Àm0%, jûPuPu B9 E5 61 F3 A4 CB DD BE 07 B1 04 Jumpu/Juó×É2 75 13 83 C5 10 E2 F4 CD 18 8B F5 Sn.j.umuMädá 19 38 2C 74 F6 A0 B5 07 B4 07 8B mElitBs, Cö Jumpu/Jaúsé2 73 2A FE 46 18 80 7E 84 08 74 08 NuÈF.s=bFUT MI 73 2A FE 46 18 80 7E 84 08 74 08 NuÈF.s=bFUT MI NuÈF.s=bFUT MI NuÈF.s=bFUT 9 08 66 63 M".tt< <td>41 41 9 18 18 14 18 14 08 14</td>	41 41 9 18 18 14 18 14 08 14	Image: Solution of the state of th
N: 0000 + 0078 33 (*3*) 1ddress Hex dunp 30869678 33 CØ 8E DØ 30869678 33 CØ 8E DØ BC 30869678 33 CØ 8E DØ BC 308696988 BF 18 06 50 57 308696988 BF A 05 9 4 30869688 FO AC CØ 9 4 30869688 FO AC CØ 9 74 308696988 FO AC CØ 9 74 308696988 FO AC CØ 9 74 30869698 FO AC AC 6 80 56 30869788 BF FE FD 30869788 AF 72 372 85 30869788 BF FE AD 808 AF 72 372	B0 7C FB 50 07 50 1F FC BE 1B 7C 3Àm0%, jûPuPu B9 E5 61 F3 A4 CB DD BE 07 B1 04 JUPuPu <	Haddress Hex dump Asscil Bits Address Hex dump Asscil Bits OBB69678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 0B 47 40 2C Asscil Asscil Bits OBB69678 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F 83 DA 30 U Asscil Asscil Bits OBB69688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F 83 DA 30 B Asscil Asscil <th< th=""></th<>	
Address Hex dunp 90869678 33 C8 8E 08 90869678 33 C8 8E 08 50 90869678 33 C8 8E 08 50 57 90869698 87 C8 08 65 57 90869698 83 66 09 74 90869688 F0 AC 30 74 90869688 B0 72 40 74 90869688 B0 72 40 74 90869788 B0 72 40 88 90869788 B0 77 72 88 90869788 G1 72 35 9	90 7C FB 50 07 50 1F FC BE 1B 7C 3ÀĐ0¼.jûPHPI 89 E5 01 F3 A4 CB DD BE 07 B 64 7 B1 B4 ;µHPW'ãuốxÉ? 75 13 83 C5 10 E2 F4 CD 18 89 F5 8n.j.uuññáč ;µHPW'ãuốxÉ? 19 38 2C 74 F6 A0 85 07 84 07 88 URAILES, to, 10 88 F5 8n.j.uuññáč 10 73 2A FE 46 10 80 7E 84 67 68 67 75 28 46 62 68 30 "T. the fluithe, to," 10 55 A0 61 67 75 28 64 62 63 30 "T. the fluithe, to," 10 10 10 10 10 10<	Image: Second state Address Hex dump Address Hex dump Address Hex dump UXal 99869678 1D C5 DD 75 CF AE A 90 A 40 2C MAVIS;	
Address Hex dunp 90869678 33 C0 8E D0 90869678 33 C0 8E D0 BC 90869678 33 C0 8E D0 BC 90869678 33 C0 8E D0 BC 90869688 BF 18 66 50 57 90869698 F0 AC 3C 00 74 908696968 F0 AC 3C 60 74 908696968 F0 AC 3C 60 74 908696968 F0 AC 3C 60 74 908696968 F0 AC 3C 74 90869678 906 74 90869678 BC 81 3E FE 70 9086978 80 83 56 90869788 B FE 72 37 20 56 90869778 GA 98	90 7C FB 50 07 50 1F FC BE 1B 7C 3ÀĐ\$¼, 1ÔPHPH 89 E5 01 F3 A4 CB DD E 7B B4 4, 1ÔPHPH 89 E5 01 F3 A4 CB BD BE 07 B1 B4 ; MHPM' 3M6xE2; 75 13 83 C5 10 E2 F4 CD 18 89 F5 8n.,]UNMÀdát 19 38 2C 74 F6 A0 85 07 84 07 80 MÆIItB, co, ju 73 2A FE 46 10 80 7E 84 67 88 MEIItB, co, ju 81 61 FE .	Image: Second state Address Hex dump Address Hex dump Address Hex dump Address Addres Address Address </th	
No. No. <th>ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3Àm0¼, 1ÔPMPH B9 E5 01 F3 A4 CB DD BE 07 B B4 3Åm0¼, 1ÔPMPH B9 E5 01 F3 A4 CB BD BE 07 B1 B4 3Åm0¼, 1ÔPMPH B7 13 83 C5 10 E2 F4 CD 18 BF S N JAMPM'ÂmôxÉ2 FC B6 07 00 B4 0E CD 18 BF S S N FULLS, C JAMPM'ÂmôxÉ2 FC B6 07 05 40 80 FE S S S T MW ME JAMPM'ÂmôxÉ2 S S A HU MW JAMPM'A MW S S A JAMPM'A S S A <</th> <th>Image: Instant instant Address Hex dump Address Hex dump Image: Instant instant 99869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 0B 47 40 2C Image: Instant instant Image: Instant</th>	ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3Àm0¼, 1ÔPMPH B9 E5 01 F3 A4 CB DD BE 07 B B4 3Åm0¼, 1ÔPMPH B9 E5 01 F3 A4 CB BD BE 07 B1 B4 3Åm0¼, 1ÔPMPH B7 13 83 C5 10 E2 F4 CD 18 BF S N JAMPM'ÂmôxÉ2 FC B6 07 00 B4 0E CD 18 BF S S N FULLS, C JAMPM'ÂmôxÉ2 FC B6 07 05 40 80 FE S S S T MW ME JAMPM'ÂmôxÉ2 S S A HU MW JAMPM'A MW S S A JAMPM'A S S A <	Image: Instant instant Address Hex dump Address Hex dump Image: Instant instant 99869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 0B 47 40 2C Image: Instant instant Image: Instant	
Normalized by the second sec	ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3Àm04.10PuP B9 E5 01 F3 A4 CB BD BE 07 B B4 7E 1B 84 7E 13 83 C5 10 E2 F4 CD 18 8E F5 8n.j.unnAnač 19 38 2C 74 F6 A0 B5 07 84 07 8B METILS,CO 1 73 28 PE 46 19 89 7E 84 07 80 67 15 73 2A FE 46 19 89 7E 84 97 80 87 10 87 10 7 11 75 5A 74 68 87 75 16 67 7 10 7 10 87 10 10 </th <th>Image: Instant State Address Hex dump Address Hex dump Address Hex dump Image: Image:</th>	Image: Instant State Address Hex dump Address Hex dump Address Hex dump Image:	
Normalized Normalin Normalized Normalize	ASCII 00 7C FB 50 07 50 1F FC BE 1B 7C 3Àmb¼, ÔPMPH B9 E5 01 F3 A4 CB BD BE 07 B B4 7E B4 B4 B4 B4 B4 C B4 B4 C Samp¼, ÔPMPH B9 E5 01 F3 A4 CB B0 BE 07 B1 B4 C Samp¼, ÔPMPH B9 25 01 F3 A4 CB B5 B7 B4 07 B8 B4 B1 B5 B7 B4 B7 B8 B6 C B1 B6 C C B7 B4 07 B8 F5 C B4 B8 C C B6 B7 UN C C C C C C C C C C C C	Image Address Hex dump Assist IIII 90869678 1D C5 DD 75 CF AE A1 9D 54 5D 92 CB 08 47 40 2C Avuïa; Tielea, IIII 90869688 E3 6F 99 81 0C A8 89 E5 CB 8A 23 2C 92 6F B3 DA Goud	

Charlen View First Accent 22 Author View First Accent 23 Accent 24 Author View First Accent 24 Accent 24 Author View Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 Accent 24 <	00401A12 C745 AC 000000 MOU DWORD PTR SS:[EBP-54],0	
UNUMERT DBM CR. (NORD / IC. S): [4872-41] DBM CR. (NORD / IC. S): [4872-41] DBM CR. (NORD / IC. S): [4872-41] SCI (BATZAH) CANDING / IC. S): [4872-41] CANDING / IC. S): [4872-41] CANDING / IC. S): [4872-41] DBM CR. (NORD / IC. S): [4872-41] SCI (BATZAH) CANDING / IC. S): [4872-41] CANDING / IC. S): [4872-41] DBM CR. (NORD / IC. S): [4872-41] DBM CR. (NORD / IC. S): [4872-41] SCI [4872-41] CANDING / IC. S): [4872-41] CANDING / IC. S): [4872-41] DBM CR. (NORD / IC. S): [4872-41] DBM CR. (NORD / IC. S): [4872-41] SCI [472-46] DB D D D D D D D D D D D D D D D D D D	00401A19 FF15 A4204100 CALL DWORD PTR DS:[4120A4]	kernel32.DeviceIoControl
0.1 1000000000000000000000000000000000000	UUUUIAIF 8840 AC MUU ECX,DWURD PIR SS:[EBP-54]	nBytesReturned> Stack SS: 9912FAB4 =9999992C
Internation OSCIL Internation OSCIL Internation 01124200 21.000 00.0	D3.[004120H4]-76001025 (Kernei32.Device1060ncr01)	
1017 FERD 20: 08 08 08 08 00 10 00 00 00 00 20 00 00 00 20 00 00 00 00	Address Hex dump	ASCII
entr2Feb0 (s 12 00 00 10 0 00 00 00 00 00 00 00 00 00 0	9912F8C9 2C 99 99 99 99 99 94 1C 99 99 99 99 99 99	9
entropy = [0] fs 30 00	0012F8D0 88 13 00 00 90 98 86 00 2C 00 00 00 2A 00 00 FF	00009*.ij 0012F8A0 0012F8C0 InBuffer = 0012F8C0
anit2Ferei an	9812F8E8 DE 38 00 00 01 00 00 00 00 00 00 00 00 00 00	9
00172 PH010 00100 PH010 00100 PH010 00100 PH0100 00100 PH01000 00100 PH01000 0010 PH01000 0010 PH01000 00100 PH01000 00100 PH01000 <t< td=""><td>0012F8F0 00 00 00 00 00 00 00 00 00 00 00 00 0</td><td>0012F8H8 0012F8C0 00LBuffer = 0012F8C0 001000000000000000000000000000000000</td></t<>	0012F8F0 00 00 00 00 00 00 00 00 00 00 00 00 0	0012F8H8 0012F8C0 00LBuffer = 0012F8C0 001000000000000000000000000000000000
00129710196 00129710196 00129710196 00129710196 00129710196 00129710000000 00129710000000 00129710000000 00129710000000 00129710000000 00129710000000 00129710000000 00129710000000 00129710000000 00129710000000 00129710000000 001297100000000 001297100000000 001297100000000 001297100000000 001297100000000 001297100000000 0012971000000000 0012971000000000000000000000000000000000		[1,,,]
Single stress and Settings\ressarch\Desktop\Sig6>DiskSector.exe Zdisk 0 Zdisk 0 Single stress 2 Settings\ressarch\Desktop\Sig6>DiskSector.exe Zdisk 0 Zdisk 0 Zdisk 0 Single stressarch\Desktop\Sig6>DiskSector.exe Zdisk 0 Settings\ressarch\Desktop\Sig6>DiskSector.exe Zdisk 0 Zdisk 0 Zdisk 0 Single stressarch\Desktop\Sig6>DiskSector.exe Zdisk 0 Zdisk 0 </td <td>0012F910 96 F9 T2 00 H8 TB 40 00 2H 00 00 00 00 00 00 00 00 00 00</td> <td>9 9012F8B4 90000000 - pOverlapped = NULL</td>	0012F910 96 F9 T2 00 H8 TB 40 00 2H 00 00 00 00 00 00 00 00 00 00	9 9012F8B4 90000000 - pOverlapped = NULL
::::::::::::::::::::::::::::::::::::		
36568 JUSplaying the data read from the sector (in hexadecimal, output redirection can also work) Sector 0x00FFDE38 (167685568 decimal) before 3.46 0x00 0x00 <td>C:\Documents and Settings\research\Desktop\i386>DiskSect</td> <td>cor.exe /disk 0 /read 167</td>	C:\Documents and Settings\research\Desktop\i386>DiskSect	cor.exe /disk 0 /read 167
Classify in the diract read in Front the sector for the metaderinal; output Path Sector 0.x00FFDE33 (16768568 decimal) before the write operation 3d 0 Md 0	18568 Displaying the data wood from the coston (in hey-decimal	autnut radiusation and
Jutput Jutput<	also work)	, output reurrection can
3:43 7:43 7:43 7:43 7:44 7:45	Dutput Data: 3x0 Ax0 Ax0 Ax0 Ax0 Ax0 Ax0 Ax0 Ax0 Ax0 A	ave eve eve eve sector 0x00EEDE38 (16768568 decimal) before
3+8 6x-0	3x8 8x8 8x8 8x8 8x8 8x8 8x8 8x8 8x8 8x8	axe axe axe axe axe sector axer pess (10,0000 decimal) perore
3/26 0x0 0x0 <td< td=""><td>1x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0</td><td></td></td<>	1x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0	
3AB BX-B	<u>3</u> ×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0	axe exe exe exe exe axe exe exe exe
2x8 bx8	3×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0	
1x8 0x8	<u>1</u> 20 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0	
3x8 9x8 0x8	1×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0	3×0 0×0 0×0 0×0 0×0 DE 38 00 00 01 00 00 00 00 00 00 00 00 00 00
3×8 0×8	<u>7</u> 20 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0	1x0 0x0 0x0 0x0 0x0 1x0 0x0 0x0 0x0
3x8 8x8	1x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0	
3x8 0x8	<u>1x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 </u>	$\frac{1}{2}$ $\frac{1}$
3x8 0x8	3x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0	3×0 3×0 3×0 3×0 3×0 3×0 DataTransferLength \rightarrow 0x 200 [13 th 16 th Bytes]
3x8 9x8	3×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0	1x0 0x0 0x0 0x0
3x8 9x8	0x0	$\mathbf{X} \mathbf{X} \mathbf{Y} \mathbf{X} \mathbf{X} \mathbf{Y} \mathbf{X} \mathbf{X} \mathbf{Y} \mathbf{X} \mathbf{Y} \mathbf{X} \mathbf{Y} \mathbf{X} \mathbf{Y} \mathbf{X} \mathbf{Y} \mathbf{Y} \mathbf{Y} \mathbf{Y} \mathbf{Y} \mathbf{Y} \mathbf{Y} Y$
3x0 0x0 0	3×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0	$0 \times 0 = 0 \times 0 = 0 \times 0$ Opcode $\rightarrow 0 \times 2 A$ [WRITE (10) command]
3x8 0x8 0	<u> </u>	IXU UXU UXU UXU UXU IXU UXU UXU UXU
3x8 8x8 8	1×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0 0×0	3x9 0x9 0x9 0x9 0x9
3x8 9x8 9	1x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0	dx0 0x0 0x0 0x0 dx0 0x0 0x0 0x0 0x0 Table 154 — WRITE (10) command
1x0 0x0 0	1x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0	
Dx8 0x8 0]x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0	0x0 0x0 0x0 0x0 0x0 Bit 7 6 5 4
3x0 0x0 0	3x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0	1x8 0x8 0x8 0x8 0x8 1x8 0x8 0x8 0x8 0x8 Byte
3x8 9x8 9	1x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0	3x0 0x0 0x0 0x0 0x0 0 0 OPERATION CODE (2Ah)
3x0 0x0 0	<u>1</u> x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0	$\frac{1}{1}$ WRPROTECT DPO
C:\Documents and Settings\research\Desktop\i386>DiskSector.exe /disk 0 /read 167 68568 3 Displaying the data read from the sector (in hexadecimal, output redirection can also work) 4 Dutput Data: 5 Bx1D 0xC5 0xDD 0x75 0xCF 0xAE 0xA1 0x9D 0x54 0x5D 0x92 0xCB 0xB 0x47 0x40 0x2C 5 0x1x0 0xC5 0xDD 0x75 0xCF 0xAE 0xA1 0x9D 0x54 0x5D 0x92 0xCB 0xB 0x47 0x40 0x2C 7 0x1x0 0xC5 0xDD 0x75 0xCF 0xAE 0xA1 0x9D 0x54 0x5D 0x92 0xCB 0xB 0x47 0x40 0x2C 7 0x1x0 0xC5 0xD0 0x75 0xCF 0xAE 0xA1 0x9D 0x54 0x5D 0x92 0xCB 0xB 0x47 0x40 0x2C 7 0x1x0 0xC5 0x30 0x9F 0x14 0x0 0x61 0x8 0x23 0x2C 0x92 0xCF 0xB3 0x13 0x9B 0x13 0x9B 0x57 0x9B 0xC6 0x80 0x9F 0x14 0x0 0x61 0x8 0x2C 0x92 0xCC 0x86 0x97 0x74 8 0x57 0x9B 0xC6 0x80 0xF0 0x84 0x52 0xEB 0xDC 0x3C 0xDA 0xD8 0xEB 0x79 0x33 0x43 0x00 0x84 0x66 0x9B 0xCC 0x8D 0x6E 0x97 0x74 0x40 0xB 0x54 0x89 0x74 0xD 0x54 0x28 0x0 0x8B 0x2E 0x7F 0x7C 0x66 0x77 0x7F 0x66 0x4D 0xB 0x15 0x10 0x85 0x26 0x28 0x0 0x8E 0x91 0x72 0x68 0x27 0x7C 0x66 0x00FFDE38 0x20 0x77 0x7F 0x66 0x19 0x32 0x22 0x28 0x0 0x8E 0x91 0x7E 0x68 0x27 0x7C 0x66 0x00FFDE38 0x40 0x8 0x16 0x7F 0x6A 0x20 0x8E 0x16 0xCE 0x8 0x16 0x7B 0x2A 0x22 0x37 0x00FFDE38 0x20 0x77 0x7F 0x66 0x19 0x32 0x22 0x28 0x0 0x8E 0x91 0x7E 0x68 0x27 0x7C 0x68 0x20 0x40 0x77 0x7F 0x66 0x19 0x8E 0x10 0x8E 0x10 0x	<u>3x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 0x0 </u>	3x0 0x0 0x0 0x0 0x0 2 (MSB) LOGICAL BLOCK ADDRE
68568 4 Displaying the data read from the sector (in hexadecimal, output redirection can also work) 5 Output Data: 6 Reserved 6 9x1D 0xC5 0xDD 0x75 0xCF 0xAE 0xA1 0x9D 0x54 0x5D 0x92 0xCB 0xB 0x47 0x40 0x2C 7 (MSB) 7 0x1D 0xC5 0xDD 0x75 0xCF 0xAE 0xA1 0x9D 0x54 0x5D 0x92 0xCB 0xB 0x47 0x40 0x2C 7 (MSB) 7 (MSB) 0x1D 0xC5 0xDD 0x75 0xCF 0xA8 0x10 0x9P 0x14 0x0 0x61 0x8 0x20 0x2C 0x92 0xCF 0xB3 0x13 0x9P 8 7 (MSB) 7 (MSB) 0x57 0x9B 0xC6 0x80 0x9F 0x14 0x0 0x61 0x8 0x6C 0x2D 0x1C 0xE2 0xCC 0x86 0x97 0x7P 9 0 8 9 0 0x57 0x9B 0xC6 0x80 0x9F 0x14 0x4 0x5E 0xBE 0xDC 0x8D 0x8E 0x9F 0x72 0xCF 0x68 0x77 0x7P 0x66 0x90 0x30 0x9F 0x14 0x40 0x6 0x0E 0x1C 0xE2 0xCC 0x86 0x97 0x7P 9 9 0 0x40 0x8 0x8P 0x33 0x43 0x00 0x84 0x66 0x9B 0xCE 0x8D 0x8E 0x9F 0x7P 0xCF 0x66 0x27 0x7P 0xCF 0x66 9 9 0 0 0x4D 0x8 0x1F 0x15 0x00 0x5A 0x28 0x0 0x8F 0x2B 0x2E 0x1P 0x72 0x68 0x27 0xCF 0x66 0x27 0x5P 0x26 0x40 0x2 0x2 0x7C 0x5B 0x26 0x20 0x2 0x00 FFDE38 0x4D 0x7 0x7F 0x6F 0x19 0x28 0x00 0x8F 0x2B 0x2E 0x2F 0x7P 0xCF 0x66 0x27 0x5P 0x2A 0x24 0x2 0x20 0x2 0x20 0x2 0x20 0x2 0x20 0	C:\Documents and Settings\research\Deskton\i386>DiskSect	cor.exe /disk Ø /read 162 3
Displaying the data read from the sector (in hexadecimal, output redirection can also work) Output Data: 8x1D 0xC5 0xDD 0x75 0xCF 0xAE 0xA1 0x9D 0x54 0x5D 0x92 0xCB 0xB 0x47 0x40 0x2C 0xE3 0x6F 0x99 0x81 0xC 0xA8 0x89 0xE5 0xCB 0x8A 0x23 0x2C 0x92 0x6F 0xB3 0x0A 0xFA 0xF5 0x81 0x76 0x30 0x9F 0x14 0x0 0x61 0x8 0x26 0x92 0x6F 0xB3 0x13 0x9B 0x57 0x9B 0xC6 0x8A 0xFA 0x47 0x4F 0x4A 0x5E 0xBE 0xDC 0x3C 0xDA 0xD8 0xEB 0x79 0x38 0xD1 0x20 0x6D 0xF0 0xB4 0x66 0x9B 0xCC 0xBC 0xBC 0x9C 0x9C 0x97 0x74 0x46 0xB9 0x33 0x43 0xD0 0xB4 0x66 0xPB 0xAE 0xAE 0xDE 0x1C 0xCE 0x86 0x97 0x74 0x4D 0xB 0xLF 0x15 0xD0 0x5A 0x28 0x0 0xFB 0xAE 0x64 0xC2 0xC2 0x66 0x27 0xFC 0x66 0xB6 0xB4 0xE9 0x90 0x5A 0x28 0x0 0xFB 0xAE 0xAE 0xD9 0x72 0x68 0x27 0xCF 0x66 0xAD 0xB 0xLF 0x15 0xD0 0x5A 0x28 0x0 0xFB 0xEB 0xD9 0x72 0x68 0x27 0xCF 0x66 0xAD 0xF 0xF9 0x90 0x8F 0xAD 0xAE 0xAE 0xEB 0xD9 0x72 0x68 0x27 0xCF 0x66 0xAD 0xF 0xF9 0x90 0x8F 0xAD 0xAE 0xAE 0xEB 0xD9 0x72 0x68 0x27 0xCF 0x66 0xAD 0xF 0xF9 0x90 0x8F 0xAD 0xAE 0xAE 0xEB 0xD9 0x72 0x68 0x27 0xCF 0x66 0xAD 0xF 0xF9 0x90 0x8F 0xAD 0xAE 0xAD 0xEB 0x9F 0xF4 0xCD 0xCF 0x8 0x27 0xCF 0x00FFDE38 RC4 Encrypted MBR written to sector 0x00FFDE38	68568	4
0utput Data: 6 Reserved GROUP NUME 0utput Data: 0x10 w07K9 0x56 0x10 0x75 0xCF 0xAE 0xA1 0x90 0x54 0x50 0x92 0xCB 0xB 0x47 0x40 0x2C 0x20 0x10 w07K9 0x81 0x75 0x30 0x9F 0x14 0x0 0x61 0x8 0x23 0x2C 0x92 0x6F 0xB3 0x10 0x99 0x65 0x10 0x26 0x80 0x97 0x14 0x0 0x61 0x8 0x2C 0x76 0x59 0x23 0x2C 0x92 0x6F 0xB3 0x10 0x99 0x13 0x98 0x57 0x9B 0xC6 0x80 0x9F 0x14 0x0 0x6F 0x80 0x22 0x2C 0x92 0x6F 0xB3 0x10 0x99 0x13 0x10 0x8F 0x4F 0x4F 0x4F 0x4F 0x4F 0x8F 0x4A 0x55 0xEE 0x1C 0x2E 0xCC 0x86 0x97 0x74 8 0x57 0x98 0x10 0x28 0x00 0x84 0x46 0x55 0xEE 0x1C 0x2E 0xCC 0x86 0x97 0x74 9 000000000000000000000000000000000000	Displaying the data read from the sector (in hexadecimal	, output redirection can 5
8x1D 0xC5 0xDD 0x75 0xCF 0xAE 0xA1 0x54 0x5D 0x92 0xCB 0xAF 0x40 0x2C 0xE3 0x6F 0x97 0x81 0x76 0xAB 0xB7 0xE5 0xE1 0xE0 0xA23 0x2C 0x92 0x6F 0xB3 0xD1 0xFA 0xB8 0xB7 0xE3 0x2C 0xP2 0xE3 0xD1 0xD3 0xD2 0xB3 0xD2 0xB3 0xD1 0xB3 0xD1 0xB8 0xP7 0xAA 0xSE 0xBE 0xD2 0xD2 0xC2 0xP2 0xC6 0xB8 0xD1 0xD3 0xD2 0xD2 0xD2 0xC1 0xB8 0xD1 0xD3 0xD2	Output Data:	6 Reserved GROUP NUME
bxcb bx7	0x1D 0xC5 0xDD 0x75 0xCF 0xAE 0xA1 0x9D 0x54 0x5D 0x92 0	0xCB 0xB 0x47 0x40 0x2C 7 (MSB) TRANSFER LENG
8x57 0x9B 0xc8 0xFA 0x47 0x8F 0x4A 0x5E 0xBE 0x1C 0x2B 0x2B 0x79 9 0x38 0x11 0x20 0x6D 0xF0 0xB4 0x6C 0xDE 0x1C 0xE2 0xCC 0x8E 0x97 0x74 9 0x08 0x07 0x74 9 0x08 0x07 0x74 9 0x08 0x07 0x74 9 0x08 0x07 0x74 0x0 0x08 0x07 0x74 0x1 0x42 0x08 0x79 0x74 0x0 0x08 0x16 0x07 0x76 0x48 0x77 0x76 0x16 0x1 0x42 0x0 0x18 0x26 0x27 0x57 0x67 0x26 0x77 0x67 0x06 0x16 0x26 0x27 0x56 0x27 0x56 0x26 0x77 0x66 0x06 0x26	0xF3 0xF5 0x81 0x76 0xD5 0x30 0x9F 0x14 0x0 0x61 0x8	0x76 0x59 0xE3 0x13 0x9B 8
bxx56 bxx6 bxx6 bxx6 bxx6 bxx6 bxx76 bxx6 bxx76 bxx77 bxx67 bxx66 bxx86 bxx77 bxx67 bxx67 bxx68 bxx77 bxx67 bxx68 bxx77 bxx68 bxx28 bxx88 bxx28 bxx88	0x57 0x9B 0xC8 0x8A 0xFA 0x47 0x8F 0x4A 0x5E 0xBE 0xDC 0	AX3C 0xDA 0xD8 0xEB 0x79
0xCD 0xFB 0x5A 0xA2 0xA1 0x4E 0x64 0xE3 0xE4 0xCD 0xCF 0x8 0x77 0xCF 0x4D 0xB 0x1F 0x15 0xD0 0x5A 0x28 0xFB 0x8E 0xD9 0x72 0x68 0x27 0xCF 0x60 0xB6 0xB4 0x15 0xD0 0x5A 0x28 0x16 0xC1 0xCB 0xB1 0xB72 0x66 0x27 0xC6 0x60 0x00 0x00FFDE38 0x47 0xC6 0x19 0xA8 0xA6 0xA6 0xA6 0xA7 0xC6 0x21 0xC0 0x37 0xC1 0x00FFDE38 0x47 0xC6 0x19 0xA8 0xA6 0xA6 <td>0x46 0xB9 0x33 0x43 0xD0 0xB4 0x86 0x9B 0xCD 0xBC 0x1C 0</td> <td>0x9F 0xF3 0x4B 0x98 0x1A</td>	0x46 0xB9 0x33 0x43 0xD0 0xB4 0x86 0x9B 0xCD 0xBC 0x1C 0	0x9F 0xF3 0x4B 0x98 0x1A
8x4D 8x8 8x1F 8x15 8xD0 8x5H 8x28 8x8 8x8 8x8E 8xD7 8x72 8x58 8x27 8xFC 8x58 8x86 8x84 8xE9 8x90 8x8F 8x6D 8xE3 8x16 8xC1 8xCB 8xB1 8xBF 8xB8 8x2A 8xC2 8x37 8x47 8xF7 8x66 8x19 8xAE 8xAD 8xA8 8x96 8xEA 8x4F 8xB5 8xE8 8x2F 8x2A 8x2A 8x24 8x28 8x69 8xC7 8x56 8x7F 8xAA 8x7 8x32 8x2F 8xAB 8x5D 8xF6 8x1 8x2A 8x2A 8x2A 8x24 8x28	0xCD 0xFB 0x5A 0x89 0x74 0xD 0xA1 0x4E 0x64 0xE3 0xE4 0	AxCD 0xCF 0x8 0x77 0xCF RC4 Encrypted MBR written to sector
8x47 0xF7 0x66 0x19 0x6E 0x6D 0x68 0x96 0xE6 0x4F 0x85 0xE0 0x7B 0x2A 0xD4 0x20 0x69 0xC7 0x5B 0x7E 0x6A 0x7 0x32 0x2E 0x6B 0x5D 0x86 0x1 0x2A 0xE1 0x80 0x1E	0x4D 0xB 0x1F 0x15 0xD0 0x5H 0x28 0x0 0xFB 0x8E 0xD9 0 0xB6 0xB4 0xE9 0x90 0x8F 0x6D 0xE3 0x16 0xC1 0xCB 0xB1 0	0.72 0.00 0.27 0.76 0.860 0.800 0.800 FFDE38
MXXX MXXX MXXXX MXXX MXXX MXXX MXXXX MXXXX MXXXX MXXX MXXX MXXXX MXXXX MXXXX	0x47 0xF7 0x66 0x19 0xAE 0xAD 0xA8 0x96 0xEA 0x4F 0xB5 0	3xE0 0x7B 0x2A 0xD4 0x20
0x10 0xC5 0x3A 0xF4 0xE 0x2C 0x3B 0x73 0x56 0x6D 0xB3 0x95 0x53 0x53 0x2 0xCE	0x57 0x57 0x58 0x7E 0xHH 0x7 0x32 0x2E 0xHB 0x5D 0xB6 0 0x10 0xC5 0x3A 0xF4 0xE 0x2C 0x3B 0x73 0x56 0x6D 0xB3 0	3x1 $0x2H$ $0xE1$ $0x80$ $0x1E0x95$ $0x53$ $0x53$ $0x2$ $0xCE$
0xFE 0xF 0xE1 0x84 0x44 0x9C 0xD5 0xFF 0xC0 0x82 0x90 0x72 0x2B 0xF5 0x9D 0xFE	0xFE 0xF 0xE1 0x84 0x44 0x9C 0xD5 0xFF 0xC0 0x82 0x90 0	0x72 0x2B 0xF5 0x9D 0xFE

(1.7 Sector where the original MBR is saved in encrypted form)

The BIOS Parameter Block (BPB) of the boot sector is then updated with information like:

- Number of Hidden Sectors in Partition [offset: 0x1C]
- Sector Number of the File System Information Sector [offset: 0x30]
- Total Sectors (in the Volume) [offset: 0x28]. This value is 1 sector less than the total number of sectors in the volume's partition table entry, because an NTFS "Backup Sector" is not considered part of the NTFS Volume.
 - Starting Cluster Number for the \$MFTMirror File in this partition [offset: 0x38].
 - This field is populated with the data which is used to identify the sector which contains primary configuration file.
 - NTFS Volume Serial Number [offset: 0x48]

•

•

• This field is populated with the decryption key.

After BPB modification, VBR is encrypted, written into the hidden file system and the storage information are saved in the secondary configuration file as shown below:





The secondary configuration file is encrypted and saved into the hidden filesystem. Information about the sector where the secondary configuration file is saved with the number of sectors and data size is stored in the primary configuration file as shown in the above picture. Later the primary configuration file is encrypted, written to the disk and storage information is saved in the malicious VBR.

The malicious VBR is written into the first (0xFFAC53) and the last sectors (0xFFFFE) of the partition.

e the boo	table in c	rder	to tr	ans	fer	cor	itro	bl t	o tł	ne	malicio	us c	ode	e as	s s	-u 00	n a	as	pos	sib	le a	as s	sho	wr	be	elo	י א(
		uau							50		PPPPPPAA		0.00		0.1												
00404BFE	8988	MOU	DAANKD Cox DM	PIK L)::[I	EHX]	,EG?	5 . 201	EG	;X=	FFFFFE80,	EHX=	0.08	028	20	<	- Ne	ΥĽ	reci	ora	upda	cea					
00404600	004624 00 0050 0h	MOU	CGA,0W	UKV 1 DTD 1	16 - 11	59.1	63F1	-00 - NY	ED	N-V	ECECCE47	05-11	ADA	000	01-	000	100	00									
00404004	885424 60	MOU	FNX NW	NRN F	TR	1-22	F (P -	FRCI		10-		1.00	0000	LUL	n] _	000	1000	000									
00404C0B	8948 08	MOU	DWORD	PTR I	S:L	EAX+	81.6	ECX	EC	X=	ØØFFAC53.	DS:F	00B6	2 B 2	E]=	000	1000	100									
00404C0E	8950 00	MOV	DWORD	PTR L	s:r	EAX+	c1.1	EDX	ED)X=	0000539D.	DS:F	00B6	2 B 3	21=	000	1000	000									
00404C11	884424 44	MOV	EAX.DW	ORD F	TR	1:22	ESP	+441			The second s	Carlo Carlo	Sector Sec		34												
00404C15	C600 00	MOV	BYTE P	TR DS	::FEI	AX],	9		DS	::r	00B62B16]	-> 08=	- \$	tat	us	of	1st	Pa	rti	tion	cha	nqei	d () b	lon-	Boo	tab	l
00404031=0	qoogleup.004	04C31			-1000				-Wester				33	C0 8	BE I	00	BC	00	7C F	B 5	0 0	7 50) 1F	FC	BE	1 B	
													BF	18 1	36 !	50	57	B9	E5 (11 F	3 A	4 CE	BD	BE	07	B1	
Address	lex dump										ASCII		38	6E (30 7	70	09	75	13 8	3 (5 1	0 E2	F4	CD	18	8B	
00B62958 3	33 CO 8E DO	BC 00	7C FB	50	97 5	0 1E	FC	BE	18	7C	3À Đ%. LÛF	PIU	83	00		49	74	19	38 2	67	4 F		1 85	07	84	07	
00B62968 E	BF 1B 06 50	57 B9	E5 01	F3 (14 C	B BD	BE	07	B1	04	; PPW'ånd	j# 1/3/4	LE I	HG 1	56 I	00	74	r 6 79	56 U 20 C		10 B	4 UL	: GD	10	EB	12	
00B62978 3	38 6E 00 7C	09 75	13 83	C5 -	10 E	2 F4	CD	18	8B	F5	8n. .um	i âôÍ	4E 80	7E 1	10 4 34 1	40	00 7.h	05	2H F 60 0	6 0	17 7	5 05	0 00	04 1.6	00	06	
00B62988 8	83 C6 10 49	74 19	38 20	74	6 A	0 B5	07	B 4	07	8B	MÆNITN8,t	tö µ∎'	46	08	36 1	83	56	GA	AA F	8 2	1 0	A 73	00	40	R6	67	
00B62998 F	F0 AC 3C 00	74 FC	BB 07	00	34 0	E CD	10	EB	F2	88	ð¬<.tü≫∎.	. Île	BC	81 :	BE I	FE	70	55	AA 7	4 1	B 8	0 7E	10	00	74	68	
00B629A8 4	4E 10 E8 46	00 73	2A FE	46	10 8	0 7E	64	ØB	74	ØB	N∎èF.s*þF		B7	07 I	B	A9	8B	FC	1E 5	7 8	B F	5 CE	BF	05	00	88	
00862988 8	80 /E 04 0C	74 05	AU 86	0/	(5 D)	2 80	46	02	50	83	Transit q	UUIF	00	B4 (38 (CD	13	Ori	dina	I ME	R			IA	DE	8A	
00802968 4	40 08 00 83	50 UH	00 E8	21	00 7	3 05	HU	BO	07	EB A R		.5	43	F7	3 1	BB	D1	{ {	gine		1.12			7	E2	39	
00002908	00 01 JE FE 07 07 FR 60	Mod	ified ME	BR					20	HU 56	-##/p}0=t#		0A (77 1	23 7	72	05	1						12	BB	00	
00B629E8 0	00 B4 08 CD	13							RA	FC	ĺlí srist	(?∎b	8B -	4E (32 1	BB	56	00	CD 1	3 7	3 5	1 4F	74	4E	32	E4	
00B62A08 4	43 F7 E3 8B	D1 86	D6 B1	06 1)2 E	E 42	F7	E2	39	56	C÷ãIÑIÖ±	IÒîB÷â	50	00 0	:D :	13	EB	E4	8A 5	6	10 G	U BE	AA	55	84	41	
00B62A18	0A 77 23 72	05 39	46 08	73	IC B	8 01	02	BB	00	7C	.w#r∎9F∎s	5 »	13	12 G		81	F 15	76	HH / RA E	5 0	10 F	0 61 0 60	01	14	28	01	
00B62A28 8	3B 4E 02 8B	56 00	CD 13	73 !	51 4	F 74	4E	32	E4	8A	ENERV.ÍBS	QOTN2	0H 01	60 C		e e R Ji	h2	9R	UH F FJi C	n 1	3 6	5 OF 1 61	73	OF	hE	75	
00B62A38 5	56 00 CD 13	EB E4	8A 56	00 (50 B	B AA	55	B 4	41	CD	V.Í∎ëä∎V.	`ȻU'	32	F4 1	IA I	56	66	CD	13 E	BI	6 6	1 F9	0 63	49	6F	76	
00B62A48 1	13 72 36 81	FB 55	AA 75	30	-6 C	1 01	74	2B	61	60	∎r6∎ûVªu(BöA∎t+	6C	69 (54 3	20	70	61	72 7	4 6	9 7	4 69	6F	6E	20	74	
00B62A58 6	5A 00 6A 00	FF 76	OA FF	76	38 6	A 00	68	00	70	6A	j.j.ÿv.ÿu	j.h.	62	6C (55 1	00	45	72	72 6	F 7	2 2	0 60	6F	61	64	69	
00802008	01 OH 10 B4	42 88	19 50	13 0	01 0 4 E	1 73	UE	41	74	UB 44	all Buol	iaasi U	67 :	20 (if i	70	65	72	61 7	4 (9 6	E 67	20	73	79	73	
00B02H78 3	02 E4 0H 50	70 /4	10 EB	20		7 UJ 0 7 E	49	UL OA	70			tion.	65	6D	30 1	4D	69	73	73 6	9 (E 6	7 20) 6F	70	65	72	
	HS address of first	absolute	sector in	partition	n→F	EFFI	• F (N	Talici	ousV	BR))	load	74	69 (δE (67	20	73	797	3 7	4 6	5 60	00	00	00	00	
00860x04 P	HS address of last	absolute	sector in i	oartition	→FF	FFF	E (M	alicio	us VE	R)		0 50	50	00	10	00	00	00	00 0		10 0		000	99	00	00	
00B6 <mark>0x08 Le</mark>	BA of first absolute	sectorin	the partiti	on $\rightarrow 5$	3AC	FF 00	(Mal	iciou	sVBF	۲)		ope	00	00 0	30	00	00	00	00 C 00 G		0 0	0 00	00 0	00	00	00	
OOB6 <mark>0x0C N</mark>	umber of sectors in	partition	→ 9D 53	00 00	⇔ (0	(1000	000 -	- (LB	A + N	umk	per of sectors in	m	00	00 0	30	00	00	20	00 C	3 9	15 A	0 00	00	00	00	80	
00B <i>č</i> first partiti	ion record) — 0x10))											91	00 1	07 I	FE	FF	FF	SF F	0	10 0	0 14	AC	FF	00	00	
00B6		بارورار											00	00	30	00	00	00	00 0	10 1	10 0	0 00	00	00	00	00	
00B62AF8	00 00 00 00	00 00	00 00	00	JO 0	U 00	00	00	00	00			00	00 1	00	00	00	00	00 0	0	0 0	0 00	00	00	00	00	
	00 00 00 00 00 04 00 07 FE	00 20	44 63	35	00 3		00	00	100	01 68	,DC5	.0	30	00 (00	00	00	00	00 C	0	10 0	0 00	00	00	00	55	
00862828	01 00 07 FE FF FF 17 FF	FF FF	53 60	FF	10 00	4 HU N 52	00	00	00	66	iiii	y .	• •							001	2FBB	8	D5EI	BES	4		
00002020 00062020		99 99	66 60	88	10 9 10 0	A 88	60	60	00	88	99=New P		TEC			H				001	2FBB	C	123	E8DC	7		
00002000	00 00 00 00	00 00	00 00	0.0	an n	0 00	00	00	55	00										001	2FBC	0	4E81	383	1		

(1.12 Partition table modified)

Finally, the malicious dropper file is deleted from the infected system to remove traces of infection. It also mark files in the %TEMP% folder for deletion on reboot and the system is rebooted.

On reboot, malicious VBR residing in the hidden file system is loaded by the MBR and control is transferred to the VBR code. It first reads the sector containing primary configuration file which is later parsed to retrieve information about the secondary configuration file. The secondary configuration file is then parsed to load sectors (block named "boot" in the secondary configuration file) containing code which is responsible for hooking Interrupt Vector Table as shown below:

EB	52	90	4E	54	46	53	20	20	20	20	00	02	08	00	00	dR	ENT	FS		ete				
00	00	00	00	00	Fð	00	00	31	50	FF	00	53	HU	FF	00				ί ες	. 5% .				
30	85	66	66	60	66	66	66	AR	21	66	88	60	66	66	9.0	0	9	- 4 -	k•					
E 6	66	66	66	ß1	66	66	66	84	BF	FR	05	6.7	80	3F	12									
			-					-		-		1	adie	ate		L D	ecr	her	on K	ey				
								6	_			-"	Iuic	ate	5 36	cioi	10	be i	eau			Mali	cious	VB
51	E3	02	EB	C9	59	57	66	61	C3	F4	EB	FD	50	62	6F	ap	.d+	YWF	a+(d	2 \ bo		Bloc	ks:	
bF	/4	66	88	99	00	មម	00	00	88	99	99	88	66	55	HH	OC							00214	
														<u> </u>	-Ind	licat	es I	whic	h bl	ock		"boo	t" —	-
															to	sear	chi	in th	e lo	aded				
															COI	nfigu	Irat	ion	file					
42	4B	46	53	00	02	00	02	00	00	00	00	00	00	00	00	BK	FS.							
AB	21	00	00	01	00	00	00	80	00	00	00	00	00	00	00	12!		!	Ç					
24	62	61	64	00	00	00	00	00	00	00	00	00	00	00	00	Şb	ad.	• • •	• • • •		De			
57	21	60	00	00	60	00	00	00	18	00	00	97	08	CH 00	HB	*1 čh	112			UK-%		nfi and	-	500
24	21	07	66	OC.	00	66	00	2F	00	00	00	FA	BE	73	64	++	TCH	ap.		0+5	190	migui	anon	100
5C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	١.					1			
C6	21	00	00	03	00	00	00	EØ	01	00	00	84	DF	5D	2E	11			a	ä].				-
Se	cto	r to	be	rea	d																			
																			124					
6 D	62	72	00	00	00	00	00	00	00	00	00	00	00	00	00	mb	r							
C7	21	00	00	01	00	00	00	00	02	00	00	10	4F	B3	52	- 11	••••	••••	••••	.0¦R				
/6	62	/2	00	00	00	00	99	00	00	00	00	00	00	00	17	VD	r			1221				
62	60	60	00	00	00	00	00	00	02	00	00	56	60	82	00	hi	a		• • • •	-***				
C9	21	00	00	01	00	00	00	25	00	00	00	9E	63	F4	86	+ 1		22.3	2	Pc(ª				
61	66	66	69	64	00	00	00	00	00	00	00	00	00	00	00	af	fid							
CA	21	00	00	01	00	00	00	04	00	00	00	6D	D7	BF	70	-†				m++	0.1			
62	6F	6F	74	00	00	00	00	00	00	00	00	00	00	00	00	bo	ot.				-			
6.9	21	60	99	93	00	00	00	UF 00	05	00	00	E3	66	83	GR	-1	499		- · · ·	psa-				
00	22	04	00	40	00	00	00	00	80	00	00	82	2F	Q1	63		100Z		Ċ.	é c	S	econd	2FV	
63	6D	64	36	34	00	00	00	00	00	00	00	00	00	00	00	cm	d64		Č.		ci	nfiqu	ration	file
46	22	00	00	39	00	00	00	00	72	00	00	D2	DC	46	D8	- F''	9		.r.,	F+				
64	62	67	33	32	00	00	00	00	00	00	00	00	00	00	00	db	g32							
53	22	00	00	OD	00	00	00	00	18	00	00	22	7F	74	24	S.,	- : :			"∎t\$				
64	62	07	30	34	00	00	00	00	00	00	មម	00	00	00	00	dD	<u>g</u> 64							
							(1.1	13 PI	CTUI	e si	now	s n	ow v	/BK	aec	laes	wn	at to	loa	a)				
		F3	BD	61	ß1	67	20	0.0	77	- r	. .	• •	ьг.	гп	40	1.4	4F	5E	Π.	a.!)	-WP	INd A	N	
		83	C4	ØE	60	89	FD	ata	fro	m	bod	ot"	cod	le			A4	61	â-	. `ë	+	YWë-=	ña.	
		E3	02	EB	C9	59	51	66	61	t	3 1	+4	EB	FD	50	60	62	72	D.	d+Y	Ifa+	(4)	br	
		00	50	64	62	67	33	32	00	5	5C (64	62	67	36	34	00	00	-1	dbg3	12.1	dbq64		
		BI	oci ' < nbr'	(s t => '' <:	Sec	e lo con Orig	ade dar gina	d: ycc d M	onfi BR	gur	atio	ont	file						T.					
		d	bg.	52	or	ap	g04	(1	15	ae	Jen	ae	nt)	<=>	ra	ike (uco	m.d	1					
						(1	.14	Pictu	ire s	shov	vs h	ow	boo	t co	de (decic	les	what	t to l	oad)				

Pointer to the Interrupt Service Routine (ISR) INT13h is replaced with an address which points to the malware's code.

Nov dword yMCC2; eax ist eax, 1m nov ax, 0m nov ax, 0m nov sscdword Ac, eax ist eax, 1m nov sscdword Ac, eax interrupt Vector Table hooked -> MTT3h ISSR repaced nov sscdword Ac, eax nov <th>nov</th> <th>eax, ss:dwo</th> <th>ra 4c</th> <th></th> <th>, heat mode for (incertapt vector fable) / or objecting</th>	nov	eax, ss:dwo	ra 4c		, heat mode for (incertapt vector fable) / or objecting
NOV aX, C5 ; LX = MR00 NOV aX, MS11; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	mov	dword_9ACC2	, eax	1	
<pre>Sh1 eray, 1mm ; tAN=HUMPACU mou ax, 1mm ; tAN=HUMPACU mou ssider, 1mm; tAN=HUMPACU merrupt Vector Table hooked => NT13h ISSR replaced Interrupt Vector Table Interrupt Vector Table Interrupt Vector Table Interrupt Vector Interrupt Vector Table Interrupt Vector Interrupt Vec</pre>	mov	ax, cs			; CS = 9AC0
mov ax, MSh : : : : : : : : : : : : : : : : : : :	shl	eax, 10h		8	; EAX=F0009AC0
nov ss:dword_4C, edX : EAX=0x000065, SS:dword_4c=[UTBALE:dword_4c] = 0F00027E6 Interrupt Vector Table hooked => WT13h ISR replaced Interrupt Vector Table hooked => WT13h ISR replaced 000 53 EF 00 F0 53 FF 00 F0 52 F	mov	ax, 085h ;	11		
Interrupt Vector Table hooked => INT13h ISR replaced 0:00 53 FF 00 60 53 FF 00 60 53 FF 00 60 53 FF 00 60 53 FF 00 F0 F0 <td>mov</td> <td>ss:dword_40</td> <td>, eax</td> <td></td> <td>; EAX=9AC000B5, SS:dword_4c=[IVTABLE:dword_4c] = 0F000E3FEh</td>	mov	ss:dword_40	, eax		; EAX=9AC000B5, SS:dword_4c=[IVTABLE:dword_4c] = 0F000E3FEh
Ditt 53 FF 00 F0 50		/	Inte	rrupt Vecto	or Table hooked => INT13h ISR replaced
PAC0::000 FA 31 C0 8E D0 8C 00 7C FB 0E 1F 0E 07 66 60 88 -1+A [Uf ² E PAC0::000 16 F6 05 36 61 13 E 00 7C E5 52 75 12 36 A1 36 7C 600. [dfk616] PAC0::000 18 07 36 66 41 42 7C 66 A3 FA 05 5E 13 36 A1 16 'C 60 116 fC 600. [dfk616] PAC0::000 18 2 76 A3 FA 05 36 66 A1 42 7C 66 A3 FA 05 56 31 ''Boot' block icode loaded in PAC0::000 16 E6 48 36 00 85 80 E1 37 88 00 56 00 FC immory which PAC0::000 16 E6 48 30 00 85 80 E1 37 88 00 50 0F immory which PAC0::000 17 E0 44 48 81 06 FA 05 0E FC 10 27 14 00 80 FC 42 74 00 immory which PAC0::0000 16 EF E 20 0F 100 FE E 20 20 52 EF 20 30 52 E immory which PAC0::0000 16 EF E 20 0F FE 20 40 95 E 40 95 22 FF 1E 12 20 0F immory which PAC0::0000 16 EF E 20 0F FE E 20 20 95 E 20 20 95 E 11 10 11 10 handler immory which PAC0::0000 06 0F 07 C BE FF 00 7C BE FF 00 7C BE FF 10 20 10 95 E immory which PAC0::0000 06 0F 07 C BE FF 20 0F 00 7C BE FF 10 20 10 95 E immory which PAC0::0000 06 0F 05 22 88 0 36 95 09 0 92 2E FF 1E 12 20 0F immory which PAC0::0000 06 0F 05 22 88 0 36 90 90 92 2E FF 1E 12 20 0F immory which I.13h hook checks which service is being requested.	000 53 010 53 020 A5 030 53 040 20 050 39	FF 00 F0 53 FF 00 F0 53 FE 00 F0 87 FF 00 F0 53 01 00 C0 40 E7 00 F0 59	FF 00 F0 FF 00 F0 E9 00 F0 FF 00 F0 F8 00 F0 F8 00 F0 F8 00 F0	53 FF (53 FF (53 FF (57 EF (41 F8 (2E E8 (90 F0 53 FF 90 F0 S .=S .=S .= Interrupt Vector Table 90 F0 53 FF 90 F0 S .=S .= Interrupt Vector Table 90 F0 53 FF 90 F0 S .=S .= Interrupt Vector Table 90 F0 53 FF 90 F0 S .=S .= .= 90 F0 53 FF 90 F0 S .=S .= .= 90 F0 53 FF 90 F0 S .= .= 90 F0 53 FF 90 F0 S .= .= 90 F0 B5 90 C0 9A 90 F0 D2 EF 90 F0 Y
T13h hook checks which service is being requested. If the service doesn't involve sector read operation nction code: 02h and 42h), it calls the original INT13h handler and transfers the control back to the call shown below: T13h hook checks which service is being requested. If the service doesn't involve sector read operation and transfers the control back to the call shown below: T13h hook checks which service is being requested. If the service Mead Sectors From Drive proves that a sector from Drive prove from the involve reading from sectors then call the INT13h Handler Doc_9ACC6:	9 A C 0 : 00 9 A C 0 : 00	00 FA 31 C0 16 FA 95 120 A3 F8 95 130 B2 7D A3 140 DB 66 89 150 FE C6 88 160 05 1E B4 170 82 46 05 180 E0 10 B8 190 E5 B5 B5 190 E3 07 B5 198 06 68 09 198 90 EA FE 198 90 EA FE 198 90 EA FE 198 90 EA FE	8E D0 BC 36 81 3E 36 66 A1 F8 05 36 1E F0 05 36 D0 05 48 8A 16 36 66 A1 B5 00 36 05 B9 04 00 7C BE 7C CB 9C E3 00 F0 05 2E 88	00 7C 0 00 7C 0 4C 7C 0 66 A1 0 84 08 0 80 E1 3 FA 05 0 4C 00 0 66 A3 0 90 F8 0 80 FC 0 2E 88 2 36 D5 0 (1.15 Ir	FB 0E 1F 0E 07 66 60 88 $\cdot 1+\ddot{A}-+. vf`\hat{e}$ EB 52 75 12 36 A1 36 70 $\cdot .6\ddot{u}$ >.]dRu.6i8] 66 A3 F4 05 EB 10 36 A1 $\dot{u}^{\circ} .6fIL f\dot{u} (.d.6i]$ 84 70 66 A3 F4 95 66 31 'Boot''block CD 13 80 56 67 82 69 95 code loaded in 3F 88 9E 01 05 66 67 ememory which Hooks IVT Hooks IVT ed 90 83 EC 9E 66 61 84 04 98 40 18 89 90 90 90 90 91 92 74 96 66 11 10 10 11 13 11 13 11 13 11 13 11 13 11 13 11 11 13 11 1
Emp ah, 2 ; Function: 02h> Read Sectors From Drive jz short loc_9ACC6 ; Function: 42h> Extended Read Sectors From Drive jz short loc_9ACC6 ; Function: 42h> Extended Read Sectors From Drive jz short loc_9ACC6 ; Function: 42h> Extended Read Sectors From Drive loc_9ACC1:	T13h ho Inction c shown !	ook checks wh code: 02h and below:	nich servio d 42h), it	ce is beir calls the	ng requested. If the service doesn't involve sector read operation original INT13h handler and transfers the control back to the call
<pre>iz short loc_9ACC6 cnp ah, 42h; 'B' ; Function:42h> Extended Read Sectors From Drive ; short loc_9ACC6 popf loc_9ACC1:</pre>	P B D	26 1			- Lupetaon Wh> Wood Voctors Lyon Drawo
<pre>inp ah, 42h; 'B' iz short loc_9ACC6 popf icc_9ACC1: jmp far ptr loc_0 inc_9ACC6: in</pre>	iz	short loc	90006		, Function. 0211> Redu Sectors From Drive
jz short loc_9ACC6 popf If the requested service is doesn't involve reading from sectors then call the INT13h Handler loc_9ACC1: Otherwise, save register information first. jmp far ptr loc_0 ; INT13h Interrupt service Routine called loc_9ACC6: ; INT13h Interrupt service Routine called mov cs:byte_9B1D2, ah ; Function mov cs:byte_9B1D3, al ; Sectors to read count mov cs:byte_9B1D5, dh ; Head popf it Head (1.16 INT13h Services to monitor) the requested service involves sector read operation, it saves information like the number of sectors to ad; sector number etc before calling the original INT13h handler. After reading the sector into memory, i ecks if it matches either of the following conditions: P E Image with IMAGE_DIRECTORY_ENTRY_EXPORT.Size == 0xFA or 0x110 o If it finds any module matching the above condition, it loads dbg32 or dbg64 (fake kdcom.or depending on the OS environment	CMD	ah, 42h ;	'B'		; Function:42h> Extended Read Sectors From Drive
popf If the requested service is doesn't involve reading from sectors then call the INT13h Handler loc_9ACC1: Otherwise, save register information first. jmp far ptr loc_0 ; INT13h Interrupt service Routine called nov cs:byte_9B102, ah ; Function mov cs:byte_9B103, al ; Sectors to read count nov cs:byte_9B105, cx ; Track & Sector mov cs:byte_9B105, dh ; Head popF (1.16 INT13h Services to monitor) the requested service involves sector read operation, it saves information like the number of sectors to nd; sector number etc before calling the original INT13h handler. After reading the sector into memory, i ecks if it matches either of the following conditions: P E Image with IMAGE_DIRECTORY_ENTRY_EXPORT.Size == 0xFA or 0x110 o If it finds any module matching the above condition, it loads dbg32 or dbg64 (fake kdcom.or depending on the OS environment	jz	short loc	98006		
loc_9ACC1: sectors then call the INT13h Handler jmp far ptr loc_0 inc_9ACC6: ; INT13h Interrupt service Routine called nov cs:byte_9B1D2, ah ; Function nov cs:byte_9B1D3, al ; Sectors to read count nov cs:byte_9B1D6, cx ; Track & Sector nov cs:byte_9B1D5, dh ; Head popt- ; INT13h Interrupt Service Routine (1.16 INT13h Services to monitor) (1.16 INT13h handler. After reading the sector into memory, i cscks if it matches either of the following conditions: • PE Image with IMAGE_DIRECTORY_ENTRY_EXPORT.Size == 0xFA or 0x110 • If it finds any module matching the above condition, it loads dbg32 or dbg64 (fake kdcom.or depending on the OS environment	popf				If the requested service is doesn't involve reading from
Ioc_9ACC6: imp far ptr loc_0 import					sectors then call the INT13h Handler
<pre>imp far ptr loc_0 ; INT13h Interrupt service Routine called ; loc_9ACC6: mov cs:byte_9B102, ah ; Function mov cs:byte_9B103, al ; Sectors to read count mov cs:word_9B106, cx ; Track & Sector mov cs:byte_9B105, dh ; Head popt pushf call dword ptr cs:loc_9ACC1+1 ; INT13h Interrupt Service Routine</pre>	TOC_A8	1001:			Otherwise, save register information first.
<pre>interverties involves sector read operation, it saves information like the number of sectors to moving the following conditions: PE Image with IMAGE_DIRECTORY_ENTRY_EXPORT.Size == 0xFA or 0x110</pre>	imn	far ntr lu	nc Ø		: INT13h Interrunt service Routine called
<pre>nov cs:byte_9B1D2, ah ; Function nov cs:byte_9B1D3, al ; Sectors to read count nov cs:word_9B1D6, cx ; Track & Sector nov cs:byte_9B1D5, dh ; Head pop+ pushf call dword ptr cs:loc_9ACC1+1 ; INT13h Interrupt Service Routine (1.16 INT13h Services to monitor)</pre>	10C_9A	1006 :			
<pre>nov cs:byte_9B1D3, al ; Sectors to read count nov cs:word_9B1D6, cx ; Track & Sector nov cs:byte_9B1D5, dh ; Head pop+ pushf call dword ptr cs:loc_9ACC1+1 ; INT13h Interrupt Service Routine (1.16 INT13h Services to monitor)</pre>	mov	cs:bute 9	B1D2, ah		: Eunction
<pre>nov cs:word_9B1D6, cx ; Track & Sector nov cs:byte_9B1D5, dh ; Head popt pushf call dword ptr cs:loc_9ACC1+1 ; INT13h Interrupt Service Routine</pre>	mov	cs:bute 9	B1D3, al		; Sectors to read count
<pre>nov cs:byte_9B1D5, dh ; Head popt pushf call dword ptr cs:loc_9ACC1+1 ; INT13h Interrupt Service Routine</pre>	mov	cs:word_9	B1D6, cx		; Track & Sector
<pre>popt pushf call dword ptr cs:loc_9ACC1+1 ; INT13h Interrupt Service Routine</pre>	mov	cs:byte_9	B1D5, dh		; Head
pusht call dword ptr cs:loc_9ACC1+1 ; INT13h Interrupt Service Routine (1.16 INT13h Services to monitor) che requested service involves sector read operation, it saves information like the number of sectors to ad; sector number etc before calling the original INT13h handler. After reading the sector into memory, i ecks if it matches either of the following conditions: PE Image with IMAGE_DIRECTORY_ENTRY_EXPORT.Size == 0xFA or 0x110 If it finds any module matching the above condition, it loads dbg32 or dbg64 (fake kdcom.or depending on the OS environment 	popt				
<pre>call dword ptr cs:loc_9ACC1+1 [; INI13h Interrupt Service Routine</pre>	pushf	and a second second			
 (1.16 INT13h Services to monitor) the requested service involves sector read operation, it saves information like the number of sectors to ad; sector number etc before calling the original INT13h handler. After reading the sector into memory, i ecks if it matches either of the following conditions: PE Image with IMAGE_DIRECTORY_ENTRY_EXPORT.Size == 0xFA or 0x110 If it finds any module matching the above condition, it loads dbg32 or dbg64 (fake kdcom.or depending on the OS environment 	call	awora ptr	CS:10C_9	ACC1+1	; INITAN INTERPUT SERVICE ROUTINE
 the requested service involves sector read operation, it saves information like the number of sectors to ad; sector number etc before calling the original INT13h handler. After reading the sector into memory, i ecks if it matches either of the following conditions: PE Image with IMAGE_DIRECTORY_ENTRY_EXPORT.Size == 0xFA or 0x110 If it finds any module matching the above condition, it loads dbg32 or dbg64 (fake kdcom.or depending on the OS environment 				(1.1	16 INT13h Services to monitor)
 If it finds any module matching the above condition, it loads dbg32 or dbg64 (fake kdcom.) depending on the OS environment 	the requ ad; sect ecks if if	lested service or number et t matches eit	e involves c before c her of the	sector r alling th followin	ead operation, it saves information like the number of sectors to le original INT13h handler. After reading the sector into memory, i lg conditions:
depending on the OS environment	• rc	Image with	IMAGE D	IRECTOR	<pre>\Y ENTRY EXPORT.Size == 0xFA or 0x110</pre>
	• ٢٢	• Image with • If it find	IMAGE_D	IRECTOF	RY_ENTRY_EXPORT.Size == 0xFA or 0x110 tching the above condition, it loads dbg32 or dbg64 (fake kdcom.)

Check Boot Configuration Data (BCD) store for BcdLibraryBoolean EmsEnabled [16000020]. If it is found, replace it with BcdOsLoaderBoolean WinPEMode [26000022]. dword ptr es:[bx], 4957534Dh ; "MSWI" CMD jz short loc 9AE2D CMD dword ptr es:[bx], 4643534Dh ; "MSCF" jnz short **PEFile** dword ptr es:[bx+3Ch], 6F63646Bh ; "kdco" стр jz short loc 9AE2D public PEFile **PEFile:** CMD word ptr es:[bx], 5A4Dh ; DOS MZ Header inz short loc 9AEAC di, es:[bx+3Ch] nov word ptr es:[bx+di], 4550h ; PE Signature CMD short loc 9AEAC inz word ptr es:[bx+di+18h], 10Bh ; Magic CMP short Check Export DataDirectory Size inz dword ptr es:[bx+di+7Ch], 8FAh ; '-' ; EXPORT_DATA_DIRECTORY.Size cmp short loc 9AEAC inz byte 9B1D4, cl nov si, 501h nov Loaded module with nov CX, 6 h mp short loc 9AEB5 Export_DATA_DIRECTORY size = 0xFA or 0x110 is checked. public Check Export DataDirectory Size Check Export DataDirectory Size: cmp dword ptr es:[bx+di+8Ch], 0FAh ; '-' ; EXPORT DATA DIRECTORY.Size jz short loc 9AEA0 dword ptr es:[bx+di+8Ch], 110h ; EXPORT DATA DIRECTORY.Size CMD jnz short loc_9AEAC (1.17 Check module with IMAGE_DIRECTORY_ENTRY_EXPORT size = 0xFA or 0x110) dword ptr es:[bx], 30303631h ; 1600 cmp jnz short loc 9AFB9 CMP dword ptr es:[bx+4], 30323030h ; 0020 => BcdLibraryBoolean EmsEnabled = 0x16000020 short loc_9AFB9 jnz dword ptr es:[bx], 30303632h ; 2600 mov dword ptr es:[bx+4], 32323030h ; 0022 => BcdOSLoaderBoolean_WinPEMode = 0x26000022 mov Windows Preinstallation Environment loc_9AFB9: dword ptr es:[bx], 1666Ch CMP short loc_9AFD7 jnz cmp dword ptr es:[bx+8], 30303631h ; 1600 jnz short loc 9AFD7 dword ptr es:[bx+8], 30303632h ; 2600 MOV loc 9AFD7: dword ptr es:[bx], 4E494D2Fh cmp ; "/MIN" short loc 9AFE9 jnz mov dword ptr es:[bx], 4D2F4E49h ; "IN/M" loc_9AFE9: dword 9B1F0, 8 cmp jnz short loc 98008 byte ptr es:[bx], OBFh cmp short loc_9800B jnz dword ptr es:[bx+1], 0C0000428h short loc_9B00B cmp jnz dword ptr es:[bx+1], 0C428h mov (1.18 Windows Pre-installation Environment)

After installing Interrupt Vector Table hook, it again reads sectors containing primary and secondary configuration file. Then it looks for a block named "**mbr**" in the secondary configuration file which contains original MBR.

Now the original MBR gets control, it loads the VBR and transfers control to it. VBR boot code loads the bootstrap code (0xF Sectors following the VBR). It first loads itself at address 0D00:0000, then following 0xF sectors are loaded in the successive memory addresses. Once all the sectors have been loaded, it transfers control to the bootstrap code as shown below:

BOOT_SECTOR:7C74 push	BOOT	SECTOR:7C74	;	
BOOT_SECTOR:7C77 push 26Ah BOOT_SECTOR:7C7A retf	BOOT	SECTOR:7C74	push	00 0 0h
BOOT_SECTOR:7C7A retf	BOOT	SECTOR:7C77	push	26Ah
	BOOT	SECTOR:7C7A	retf	www.enders.

(1.19 Control transferred to bootstrap code)

Bootstrap code read contents from the root drive, loads NTLDR at address 2000:0000h and transfers control to the NTLDR.

debug004:0485 push	1 2000h
debug004:0488 pust	n ax
debug004:0489 retf	F.
(1.20 Control transferre	ed to NTLDR)

NTLDR contains an embedded PE file (osloader.exe) which loads the Windows system files (starting with the ntoskrnl.exe, its dependencies (HAL.dll, bootvid.dll, and kdcom.dll), SYSTEM hive, and the boot drivers) into memory.

osloader.exe mostly executes in protected mode, but for input/output operation it depends on the BIOS services. So it keeps switching between real and protected mode.

Since the size of kdocm.dll's EXPORT_DATA_DIRECTORY is 0xFA. As soon as kdcom.dll is loaded, INT13h hook loads fake kdcom.dll (dbg32 or dbg64) into memory, updates checksum and replaces the original kdcom.dll with the fake kdcom.dll as shown below. Control is then transferred back to the osloader.exe

kdcom.dll				
Member	Offset	Size	Value	Section
Export Directory RVA	00000150	Dword	00001300	.edata
Export Directory Size	00000154	Dword	000000FA	

(1.21 kdcom.dll IMAGE_DIRECTORY_ENTRY_EXPORT.Size = 0xFA)

MOV	di,	bx						; [)X=0																	F	ake kd	com	Original kdc	com
NOVZX	CX,	byte_	7 B 1D4	F.				; t	iyte_	9B1)4 =	ØEh																	1	
shl	CX,	9														201.0681							Nun	berd	fSect	ions	0003		0007	
cmp	CX,	word	otr d	word	_9B1F	0		; (:X=10	;00,	word	i ptr	dword_!)B1F0 = 1	1400 <	Siz	eofIna	ge					Offs	et PF	Sign	ature	D8		C8	
jle	shor	t loc	9AF8	5																			Add	Taeel	fEntry	Poin	00000	000	000007	CE6
MOV	CX,	word j	otr d	word	_9B1F	0		; 1	iord	ptr	dwor	'd_9B'	1F0 = 11	100 < 1	sizeof	Image							Siz	ofin	nada	i viii	00001	100	000015	RRA
																							5121	5 VI II	naye		00001	AUU	000011	000
loc_9AF8	35:							; (ODE	XREF	: de	ebugØ	03:037F	ľj																
MOV	si,	OB3Ah					_	_	_	_	_																			
rep movs	50		_	_	_	_		;(opy	mali	ciou	us kdo	com.dll	to the i	ienory	addre	ss whe	re th	e ori	gina	l kdco) m. dl.	1							
22F3:D0D	0 4	D 5A 9	0 00	03	00 00	00	64	00 (0 00	FF	FF Ø	00 00	MZE			22E3:	DADA	4D 5A	90.0	0 03	AA DI		64 6	0 00	88 F	FF FF	66 60	MZ	É	
22F3:D0E	0 B	8 00 0	0			-	DIL		°	00	00 0	00 00	+			22F3:	DBEB	B8 88			-	-	-	A 88	00 1	00 00	00 00	+.		
22F3:D0F	0 0	0 00 0	lo Or	gina	KDC	UNI.I	ULL		30	00	00 0	00 00				22F3:	DOFO	00 00	Mali	cious	KDC	QM.D	LL	00	00 1	00 00	00 00			
22F3:D10	0 0	0 00 0	0						30	1 D8	00 0	00 00		+		22F3:	DIAA	AA AA						88	88 (18 AG	00 00			
22F3:D11	0 0	E 1F E	A ØE	00	84 09	CD	21	B8 (1 40	CD :	21 5	4 68		!+.L-!	Th	22F3 ·	D118	SE 1F	RA Ø	F AA	R4 Ø	0.0	21 B	8 81	40. 0	CD 21	54 68		1.1++.1-	1Th
22F3:D12	0 6	9 73 2	0 70	72	6F 67	72	61	6D 2	0 63	61	6E 6	E 6F	is pro	lgram car	ino	22F3:	0120	69 73	20 7	0 72	6F 6	7 72	61 6	D 20	63 /	51 6F	6F 6F	is	program ca	nnn
22F3:D13	0 7	4 20 6	2 65	20	72 75	6E	20	69 6	E 20	j 44	4F 5	3 20	t be r	un in DC	IS	22F3:	0130	74 20	62 6	5 20	72 7	5 6F	20 6	9 6F	20 1	44 4F	53 21	i t	he run in D	20
22F3:D14	0 6	D 6F 6	4 65	2E	OD OD	ØA	24	00 (0 00	00	00 0	00 00	node	\$		22F3:	0140	6D 6F	64 6	5 2F	AD AL) RA	24 6	0 00	00 1	na ac	00 00		de S.	
22F3:D15	0 5	3 A9 C	6 5D	17	C8 B8	ØE	17	C8 P	8 ØE	. 17	C8 B	18 ØE	S-+1.+	+ ++	++.	22F3	0150	01 FL	04 6	F DS	85 E	30	05 8	5 FA	30 1	15 85	FA 31	295	ön+à•=+à•=+	à-=
22F3:D16	0 1	7 C8 E	9 ØE	18	C8 B8	ØE	ED	EB P	1 OE	. 12	C8 B	8 ØE	.+1	+.fdí+	++.	22F3	0168	N5 85	FR 3	n cc	85 F	30	DC F	D 69	30 [16 85	FA 30	+à	u=1à+= 2i=+	à.=
22F3:D17	0 E	D EB 8	7 ØE	16	C8 B8	ØE	80	EB F	D ØE	. 16	C8 B	18 ØE	fdç	+.Cd2	++.	22F3	0170	CF 18	54 3		85 F	30	CE 1	8 61	30 [14 85	FA 30	+	T=!à·=+ a=+	3.=
22F3:D18	10 C	D EB A	4 ØE	13	C8 B8	ØE	ED	EB 8	5 OE	16	C8 B	18 ØE	-dñ•	+.fdà+	H.	22F3	D180	CF 18	67 3	ID D4	85 FI	30	52 6	9 63	68 1	5 85	FA 30	+	n=+à·=Rich+	à.=
22F3:D19	0 5	2 69 6	3 68	17	C8 B8	ØE	00	00 0	0 00	00	00 0	00 00	Rich.	+		22F3	0198	00 00	00 0	10 00	66 6	3 66	50 4	5 88	00 1	LC 01	03 00	1	PF I	
22F3:D1A	0 0	0 00 0	0 00	00	00 00	00	50	45 8	0 00	j 40	01	7 00		PEL		22F3:	DIAR	F3 94	A1 4	F AA	00 0	3 66	00 0	a aa	00 1	FA AC	82 21	nö	íN a	
22F3:D1B	0 4	6 83 7	D 38	00	00 00	00	00	00 0	0 00	E O	00 0	E 21	Fâ};	a		22F3:	DIRA	AR A1	GA G	10 00	16 0	3 66	80 0	1 00	00 1	na ac	00 00	1	C	
22F3:D1C	0 0	B 01 (7 00	00	11 00	00	80	07 0	0 00	00	00 0	00 00				22F3	0100	00 00	00 0	10 80	62 6	3 00	88 1	8 88	00 1	na ac	00 10		C C	
22F3:D1D	I E	6 OC (0 00	00	03 00	00	00	0D (0 00	00	00 8	1 80	μ		C	22F3	0100	80 08	00 0	10 80	00 0	3 88	05 0	8 81	00 1	00 00	00 00	i c	C.	
22F3:D1E	0 8	0 00 0	0 00	80	00 00	00	05	00 0	1 00	05	00 8	1 00	ÇC.			22F3	DIFA	A5 AA	A1 A	ia aa	00 0	3 00	AA 1	A GO	00 5	80 02	66 66	3.	C. S. C.	
22F3:D1F	0 0	5 00 0	1 00	00	00 00	00	80	1B (0 00	00	03 0	00 00		Ç		22F3	DIEA	11 83	aa a	10 01	AA 4	1 05	88 8	A 10	00 0	AA 10		3	Q	
22F3:D20	10 1	1 83 (0 00	01	00 00	00	00	00 0	4 00	00	10 0	00 00	.â			22F3	0200	00 00	10 0	ia aa	10 0	1 00	88 6	A AA	00 -	10 00				
22F3:D21	0 0	0 00 1	0 00	00	10 00	00	00	00 0	0 00	10	00 0	10 00				22F3	D21A	30 17	00 0	A FA	00 0	1 00	10 1	5 88	00	28 80		0		
22F3:D22	0 0	0 13 (0 00	FA	00 00	00	00	14 8	0 00	30	00 8	00 00				22F3	D220	AA AA	00 0	ia aa	AA A	1 00	88 8	A AA	00 1	na ac				
22F3:D23	0 0	0 16 0	0 00	FØ	03 00	00	00	00 0	0 00	00	00 8	00 00	=.			22F3	0230	00 00	00 0	10 00	00 0	1 00	00 1	0 00	00 0	10 00	00 00		á	
22F3:D24	0 0	0 00 0	0 00	00	00 00	00	00	14 6	0 00	20	01 0	00 00				22F3.	0200	00 00	00 0	10 00	00 0	3 00	00 0	0 00	00 1	00 00	00 00			
22F3:D25	0 5	0 03 0	0 00	10	00 00	00	00	00 6	0 00	00	00 6	00 00	P			22F3.	0240	00 00	00 0	0 00	00 0	3 00	88 8	0 00	00 0	00 00	88 80			••••
22F3:D26	0 0	0 00 0	0 00	00	00 00	00	00	00 0	10 00	00	00 0	00 00				2210.	0250	00 00	00 0	0 00	00 0	3 00	88 8	0 00	00 0	00 00	00 00			
22F3:D27	0 0	0 00 0	0 00	00	00 00	00	00	00 0	10 00	00	00 0	00 00				22F3.	0200	80 02	00 0	10 50	00 0	3 00	88 8	0 00	00 0	00 00	88 80	6	X	
22F3:D28	0 0	0 03 0	0 00	44	00 00	00	00	00 0	10 00	00	00 0	0 00	D.			2210.	0200	00 02	88 A	18 8A	00 0	3 88	88 8	8 88	00 0	00 00	88 A	y.		
22F3:D29	0 0	0 00 0	0 00	00	00 00	00	00	00 6	10 0P	00	00 0	0 00				2210.	0200	9F 7h	65 7	Q 71	00 0	3 00	00 0	5 88	00 0	20 00	88 80	+	avt - P	
22F3:D2A	0 2	E 74 6	5 78	74	00 00	00	FE	09 P	10 01	00	03 P	0 00	.text			2210.	0220	00 14	00 1	0 74 18 88	00 0	3 00	00 0	0 00	00 0	00 00	00 00		C	
and wreath																2213.	VZHU	00 10	00 0	10 00	02 0	0 00	00 0	0 00	00 1	00 06	00 00		···à·····	

(1.22 Original kdcom.dll is replaced in memory with dbg32 – fake kdcom.dll)

KDCOM.DLL is COM-based debugging plug-in, so by faking the exported APIs it is actually disabling kernel debugging option via COM port.

KdD0Trans→ 36058018001001	mov	b.[010001880].1		
.100005AD: 33C0	XOP	eax,eax		
.100005AF: C3	retn			
KdD3Transition:	mov	b,[010001880],2		
.100005B7: 33C0	xor	eax,eax		
.100005B9: C3	retn			
KdDebuggerInitialize0	mov	b,[010001880],3		
.100005C1: 33C0	×012	eax,eax		
.100005C3: C20400		4		
KdDebuggerInitialize1:	push	010000602 +2		
100005CB: FF1584020010	call	PsSetCreateThreadNotifyRoutine		
.100005D1: C20400	retn	4		
KdReceivePacket:	MOV	D, LOIOOOI8801,4		
.100005DB: 33C0	xor	eax,eax		
.100005DD: C21400	retn	00014		
KdRestore: C6058018001005	mov	b,[010001880],5		
.100005E7: 33C0	xor	eax,eax	antiaco contrato co	
.100005E9: C20400	retn	4	Fake APIs and	
KdSave: C6058018001006	mov	b,[010001880],6	Poturp cuococc	
.100005F3: 33C0	×01°	eax,eax	return success	
.100005F5: C20400	retn	4 echiomannoacomes		
KdSendPacket :	mov	b,[010001880],7		
100005FF: C21000	wetn	00010		

(1.23 APIs faked by malicious kdcom.dll)

Mitigation

- Block access to the unused ports and block the access to the above mentioned URLs.
- Users who are identified to be infected are requested to change their passwords.

• Reboot the system in safe mode and log in as the Administrator user.

Execute the CSSCAN command line tool using the Beta DATs to remove any Trojan or infected file from the system:

o VSE 8.7

"C:\Program Files\McAfee\VirusScan Enterprise\csscan.exe" -All -Unzip -Program -Analyze -Sub -Clean -Log c:\scan-rpt.txt C:\

o VSE 8.8

"C:\Program Files\Common Files\McAfee\SystemCore\csscan.exe" -All -Unzip -Program - Analyze -Sub -Clean -Log c:\scan-rpt.txt C:\

• Other McAfee product users

Please use the following <u>Stinger</u> standalone tool.

To use the Stinger tool, please make sure the targets "Processes" and "Registry" are disabled and the interface "List of all files scanned" is enabled in the stinger before scanning the infected machine.

- Read more about using the Stinger tool <u>here</u>.
- Reboot the system normally.
- Run GMER again to confirm that no malicious threads of patched files exist anymore.

Restart Mechanism

The malware restarts by randomly infecting a system driver (usually located in %windir%/system32/drivers). This particular variant mostly infects the file VOLSNAP.SYS

Getting Help from the McAfee Foundstone Services team

This document is intended to provide a summary of current intelligence and best practices to ensure the highest level of protection from your McAfee security solution. The McAfee Foundstone Services team offers a full range of strategic and technical consulting services that can further help to ensure you identify security risk and build effective solutions to remediate security vulnerabilities.

You can reach them here: <u>https://secure.mcafee.com/apps/services/services-contact.aspx</u>

© 2011 McAfee, Inc. All rights reserved.