

A Sophisticated Fraudster's Rootkit





Agenda

- What are rootkits?
- TDL4
- System Infection
- Anti-rootkits
- Conclusion









A rootkit is

"A set of programs and code that allows a permanent or consistent, undetectable presence on a computer" ^[1]









 \mathbb{N}



				R			/
72	5	S.			F		?
Z . 8 -	AL	6_6-	11	AL		End of a	/

Theories for selfreplicating programs are created First Apple virus found "in the wild" - Spreads through pirated computer

games

Boot sector Virus - command com infector
 Polymorphic Virus
 Macro Virus

 Java
 Java

 - toolkits for creating viruses
 infectors

 become
 -destructive viruses that online

 online
 destroy dat on certain dates

Macro VirusMelJava-EminfectorsspaChernobyl- us-destructivedocviruses thatdocdestroy dataILoron certainvirudatesSen

Melissa -Email spammer - uses MS Word documents ILoveYou virus Sends via email

Slammer Worm - fastest spreading worm to date; infecting 75,000 computers in approximately ten minutes

TDL
Stuxnet
Rustock
Trojans
Mobile
trojans

Conficker

number of

computers

Slammer in

infected since

Worm

- Most

2003



An exponential growth on malware technology throughout the years.







In fact, there is no reason for them not to...

As rootkits provides the most basic yet most effective feature a malware needs...

Stealth





How they avoid detection

- Replacing / Modifying system files
- Modification of internal Windows structures
- Direct Kernel Object Manipulation
- Memory hooking (Shadow Walker)









Commonly "hooked" system tables

- SSDT System Service Dispatch Table
- IDT Interrupt Descriptor Table
- GDT Global Descriptor Table

	Levels of hi	di	ng files on V	Vi	indows NT
Windows application	Some Function				
Kernel32.dll	FindFirstFile/ FindNextFile	1	Import table patching		Function address modification. Function located in dynamic load library is accessible by the address in Import table. So one can intercept its call by modifying this address.
Ntdll.dll	NtQueryDirectoryFile	2	Function splicing		Modification of the function code to intercept calls. For example, one can replace the first bytes by JUMP command leading to the interceptor code.
User mode (ring 3)	Int 2Eh (sysenter for XP and later) rupt	3	GDT Global Descriptor table		CPU implementation of ring0 and ring3 levels. Using GDT we can perform operations with kernel mode privileges (ring 0) from user-mode (ring 3).
Kemel mode (ring 0) Processor	Interrupt dispatch	4	MSRs Processor Machine State Registers		MSRs are used to store handle of sysenter instruction. Manipulations with them give a possibility to set a single hook for all system calls.
	(4)	5	IDT Interrupt Descriptor Table] [Table of handlers of all interrupts. For example, the hook of int 2Eh gives a possibility to intercept all system calls.
Ntoskrnl.exe	KiSystemService KiFastCallEntry	6	SSDT System Service Descriptor Table]	SSDT contains the table of pointers for each system call. So here one has a possibility to intercept calls.
Ntoskrnl.exe	5 NtQueryDirectoryFile	7	DKOM Direct Kernel Object Modification		Modification of the internal system data to change the original behavior. For example, one can change object callbacks to intercept the creation of file objects.
	6	8	Hook KiSwapContext		KiSwapContext is called every time when processor switches the thread. For example, one can find the NKQueryDirectoryFile call located in stack and modify the return address in stack - thus the call will be intercepted.
File System	IRP_MJ_QUERY_INFORMATION	9	MJ function hook		At this level we can modify the information returned by the file system driver.
	7	10	FS filter driver		Intormation level: file records. For example, one can cut some files from the list.
Device Driver	IRP_MJ_READ	11	MJ function hook		Logic is the same as in the hook described above but here it uses the information of different level.
		12	Disk filter driver		For example, one can parse memory and hide a file by modifying FS raw data.













TDL 4

"One of the stealthiest rootkit in the wild" "Uses sophisticated techniques to avoid detection"

"32bit and 64bit support"

"Owns a very large botnet"

"Continously evolving"

C





G



Evolution of TDL

	TDL1	TDL2	TDL3	TDL4
RC4 encryption	no	no	YES	YES
Hooked Functions	NtFlushInstructionCache NtEnumerateKey PsLoadedModuleList	lofCallDriver lofCompleteRequest NtSaveKey NtQueryValueKey	AddPrintProcessor AddPrintProvidor	AddPrintProvidor ZwConnectPort
Encrypted File System	no	no	YES	YES
Priveledge Escalation	no	no	no	YES
64 bit support	no	no	no	YES
Overwriting of MBR	no	no	no	YES
Complexity (1-5)	2	2	3	4

Hiding of TDL Registries Hiding of TDL Files Hiding TCP ports Process injection





e driver



Key Features of TDL4

- Bypasses Windows *PatchGuard* for 64bit mode support
- Exploits MS10-092 for privilege escalation
- Infects the MBR to ensure survivability upon reboot
- Loaded before the operating system can initialize
- Hiding the critical files & registry keys it uses
- Creates own file system to store root
- Injecting malicious code into system
- Disables kernel debugger to avoid k
- Uses a watchdog to guard the syste
- Able to re-infect if a discrepancy is a

MS10-092

Task Scheduler Vulnerability that could allow for elevation of privilege

Task Scheduler improperly validates whether scheduled tasks run within the intended security context







Hex view of the last sector of an infected hard disk

- creates a hidden and encrypted(RC4 algorithm) partition in the last sector of the hard disk

-Using its own file system, it saves other rootkit components and the original MBR for later use.



вk

110

ту



Module	Description
mbr	Original MBR of infected system
ldr16	16bit real mode loader code
ldr32	fake kdcom.dll for 32 bit systems
ldr64	fake kdcom.dll for 64 bit systems
drv32	rootkit driver for 32 bit systems
drv64	rootkit driver for 64 bit systems
cmd.dll	payload for 32 bit processes
cmd64.dll	payload for 64 bit processes
cfg.ini	configuration
bckfg.tmp	encrypted list of C&C URLs















TDL 4 Unpack and Initialize What bit is running proces s 32bit infection 64bit infection







64bit, Improved defence?

- Code Integrity Policy prevents unsigned kernel-mode drivers on loading
- Windows PatchGuard protects modification of
 - SSDT System Service Dispatch Table
 - IDT Interrupt Descriptor Table
 - Global Descriptor Table
 - Patching codes on kernel









"The Master Boot Record (MBR) is the first 512 bytes of a data storage device that contains code for bootstrapping an operating system. It houses the table of primary partitions using the IBM partition table heme. It's primary purpose is to load the boot sector and pass control to it (volume boot record)"



			Suucluie	or a master boot record	
	Address		_	Size in	
Hex	Oct	Dec		-	bytes
0000	0000	0	code area		440 (max. 446)
01B8	0670	440	disk signatur	e (optional)	4
01BC	0674	444	Usually nulls;	2	
01BE	0676	<mark>446</mark>	Table of prin (Four 16-byte scheme)	nary partitions entries, IBM partition table	64
O1FE	0776	510	55h	MBR signature;	2
01FF	0777	511	AAh	0xAA55	2
		M	BR, total size	: 446 + 64 + 2 =	<mark>512</mark>

Structure of a master boot record











IPRE

ANTIVIRUS





PAYLOAD

- TDL "Trojan Downloader"
 - Keyloggers
 - Clickers
 - Fake AVs
 - Adware
- Receives commands from botnet C&C and runs them
- Intercepts user searches and spoofs the search result
- Creates search requests to popular search engines.









Should we be alarmed?

"Stealth is nothing new to the anti-virus industry"

























- Rootkit detection and removal is an integral part of Advanced Malware Detection
- More and more companies are opting for this technology



















🕼 Windows XP Professional - VMware Workstation 🕞	🖸 Windows XP Professional - VMware Workstation
File Edit View VM Team Windows Help	File Edit View VM Team Windows Help
Hie Edit Search View Analysis Extras Window ?	File Edit Search View Analysis Extras Window ?
📄 🚵 🗸 💭 🥥 🐉 🛃 16 🔹 ANSI 🔹 hex 🔹 🛛 🖌 🖌 Sector 0 🗘 of 8388576	🗋 🚵 🖝 🔐 🥶 16 🔹 ANSI 🔹 hex 🔹 🛛 🕯 ANSI 🔹 bector 0 🗘 of 8388576
J Hard Disk 1	J Hard Disk 1
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 33 CO 8E DO BC 00 7C FB 50 07 50 1F FC BE 1B 7C SAZD4. UP.P.W4. Sector 0	00000000 β3 CO 8E DO BC 00 7C 8E CO 8E D8 BE 00 7C BF 00 βλŽĐ4. ŽλŽØ4. ζ. Sector 0
00000010 BF 1B 06 50 57 B9 E5 01 F3 A4 CB BD BE 07 B1 04 ¿ PW*A.o×E***.±.	00000010 06 B9 00 02 FC F3 A4 50 68 1C 06 CB FB 60 B9 47úó¤Ph. Eú`'G
00000020 38 6E 00 7C 09 75 13 83 C5 10 E2 F4 CD 18 8B F5 8n. .u.fÅ.åôÍ.<ŏ	00000020 01 BD 2A 06 D2 4E 00 45 E2 FA 44 85 2E 70 1C B8 .½*.ÔN.EáúDp.,
00000030 83 C6 10 49 74 19 38 2C 74 F6 A0 B5 07 B4 07 8B f#.It.8,tö µ.'.<	00000030 26 04 08 68 62 40 0E 83 0C A3 3A 81 96 84 F5 12 &hb@.f.£:-~~ö.
00000040 F0 AC 3C 00 74 FC BB 07 00 B4 0E CD 10 EB F2 88 8-<.tu»'.I.ëô^	00000040 10 C7 03 21 01 E1 00 37 26 BF B5 C1 37 60 00 Å3 .ζ.!.ά.7είμΑ7£
00000050 4E 10 E8 46 00 73 2Å FE 46 10 80 7E 04 0B 74 0B N.éF.s*bF.€~t.	00000050 BF 00 33 E2 88 41 FF D8 E8 06 83 4C FF 8E B0 00 ¿.3& Ayde.fLyZ°.
00000060 80 7E 04 0C 74 05 A0 86 07 75 D2 80 46 02 06 83 €~t. ¶.u0€Ff	00000060 7D A8 04 E2 C1 4A 40 CF 49 A1 53 02 B0 0C AB B7 } .aJg[I];S«
00000000 46 00 05 05 56 04 00 E6 21 00 75 10 00 74 C8 01 E5 1. JVE.S. N.E	
DODDODOD BC OF SE FE / p SS XX / T OD OD / E SO OD / C CS XU π , p / p / τ , e^{π} , E	
00000010 00 B4 0B CD 13 72 23 83 C1 24 3F 98 83 DF 83 FC 1 f ##6502.55	
000000B0 43 F7 E3 8B D1 86 D6 B1 06 D2 EE 42 F7 E2 39 56 C+8(N+0+.01B+A9V	000000B0 E8 08 45 85 2E 70 6E 4C 75 04 02 AF 8F 80 89 03 è.E. pnLu
00000000 0A 77 23 72 05 39 46 08 73 1C B8 01 02 BB 00 7C .v#r.9F.s».	000000C0 62 FF 98 F6 11 F9 93 1C FD C3 BA 3E 51 F8 93 1C bt*6.ù*.tŰ>00*.
00000000 88 4E 02 8B 56 00 CD 13 73 51 4F 74 4E 32 E4 8A <n.<v.í.sqotn2aš< th=""><th>00000000 04 04 00 32 31 7C 54 B6 E4 07 44 EB 4E 70 44 3621 TTA.DENpD6</th></n.<v.í.sqotn2aš<>	00000000 04 04 00 32 31 7C 54 B6 E4 07 44 EB 4E 70 44 3621 TTA.DENpD6
000000E0 56 00 CD 13 EB E4 8A 56 00 60 BB AA 55 B4 41 CD V.Í.ëašV.'»"U'AÍ	000000E0 E4 07 23 BF D9 57 20 A4 AD 88 6B BF 78 57 EE 22 ä.#¿ÙW ×-`k¿×Wi"
000000F0 13 72 36 81 FB 55 AA 75 30 F6 C1 01 74 2B 61 60 .r6.ûU ² uOöá.t+a	000000F0 3C 72 04 22 C3 37 40 FA 45 08 5D 00 40 F0 B5 78 <r."ä7@úe.].@бµх< th=""></r."ä7@úe.].@бµх<>
00000100 6A 00 6A 00 FF 76 0A FF 76 08 6A 00 68 00 7C 6A j.j.ÿv.ÿv.j.h. j	00000100 E4 08 87 AD C0 37 40 C4 FF FE E1 A2 F1 27 38 00 ä.≠-À7@Äypácñ'8.
00000110 01 6A 10 B4 42 8B F4 CD 13 61 61 73 0E 4F 74 0B .j.'B<ôÍ.aas.Ot.	00000110 91 89 E3 A2 E5 27 38 22 1B 72 83 22 F5 27 38 00 \%ãoµ'8".rf"õ'8.
00000120 32 E4 8A 56 00 CD 13 EB D6 61 F9 C3 49 6E 76 61 2äŠV.Í.ëÖaùÄInva	00000120 D3 30 F6 62 F9 A8 6C C9 0E 30 06 91 49 57 D6 85 ÓOöbù"lÉ.O.'IWÖ…
00000130 6C 69 64 20 70 61 72 74 69 74 69 6F 6E 20 74 61 lid partition ta	00000130 87 66 98 30 1D F3 FF 06 7C A2 04 D0 88 47 68 99 +f~0.óÿ. 0.D^Gh™
00000140 62 6C 65 00 45 72 72 6F 72 20 6C 6F 61 64 69 6E ble.Error loadin	00000140 FF 0E 4A 02 CC 38 F0 62 10 00 F5 79 31 DC DD 40 ÿ.J.I88b8y1UY0
00000150 67 20 6F 70 65 72 61 74 69 6E 67 20 73 79 73 74 g operating syst	00000150 00 BE 54 02 7E 6A 49 BB 0B C9 3A 83 20 EC 1C 1B .%T.~jI».E:f 1
00000150 55 50 00 40 59 73 73 59 56 57 20 57 70 52 72 51 em. Alssing opera	00000160 20 29 F4 40 F9 98 4F 2D EA EA ET 18 8C 27 89 D8 jogu 0-eea. (F %)
000001/0 /4 69 62 67 20 73 79 73 74 65 60 00 00 00 00 00 00 Cing system	00000170 00 09 02 07 20 73 74 75 00 00 00 00 00 00 00
Offset: 0 Readonly Overwrite	Offset: 0 Readonly Overwrite
💐 Start 🛛 🚜 Install Watch Pro 2.5c - [🕪 HxD	👌 Start 🛛 💐 InstaliWatch Pro 2.5c - [🛛 🕪 HxD
The installed version of VMware Tools is not up to date. Log in to the guest operating system and Update Tools Remind Me Later Neve click Update Tools.	The installed version of VMware Tools is not up to date. Log in to the guest operating system and Update Tools Remind Me Later Never Remind Me click Update Tools.
To direct input to this VM, move the mouse pointer inside or press Ctrl+G. 🚗 💿 💾	To direct input to this VM, click inside or press Ctrl+G. 🚨 💿 💾 🔯 🖉



IPRE

ANTIVIRUS

Clean

Infected







GFI®

ANTIVIRUS

seg000:002A	MOV	ds:7B2h, dl ; save drive index
seg000:002E	sub	word ptr ds:413h, 10h ; reserve 16kb
seg000:0033	mov	ax, ds:413h
seg000:0036	shl	ax, 6
seg000:0039	mov	ds:674h, ax ; fix jump address to the buffer with ldr16
seq000:003C	mov	ah, 48h ; 'H' ; Extended Read Drive Perameters
seq000:003E	mov	si, 8C5h
seq000:0041	mov	word ptr ds:8C5h, 1Eh ; size of buffer
seq000:0047	int	13h ; DISK -
seq000:0049		all approximate a state of the second s
seq000:004C		
seq000:004C loc 4C:		; DATA XREF: seq000:00471r
seq000:004C		, neaunna ecryptBuf+421r
seq000:004C	mov	cx, 6
seq000:004F	call	FindLdr16
seq000:0052	mov	eax, [si+14h]
seq000:0056	push	word ptr ds:674h
seq000:005A	рор	es
seq000:005B	xor	di, di
seq000:005D		
seq000:005D loc 5D:		; CODE XREF: seq000:006E1j
seq000:005D	call	ReadAndDecruptBuf
seq000:0060	mov	si, 8E9h
seq000:0063	mov	cx, ds:8E5h
seq000:0067	rep move	50
seq000:0069	mov	ax, ds:8E7h
seq000:006C	test	ax, ax
seq000:006E	inz	short loc 5D
seq000:0070	popa	-
seq000:0071	imp	far ptr loc 0 : jump to ldr16 entru point
seg 000.0076	1.4	the first of the second second beauty

After decryption and loading LDR16

seq000:003F HookedInt13h: seg000:003F seq000:003F 9C seq000:0040 80 FC 02 seq000:0043 74 0B seg000:0045 80 FC 42 seq000:0048 74 06 seq000:004A 9D seq000:004B seg000:004B jmpOrig13h: seq000:004B seq000:004B EA 00 00 00 00 seg000:0050 seq000:0050 seg000:0050 ifRead: seq000:0050 seg000:0050 2E 88 26 E3 03 seq000:0055 2E A2 E1 03 seq000:0059 2E 66 C7 06 DD 03 00 00 00 00 seg000:0063 2E 66 C7 06 D9 03 00 00 00 00 seg000:006D 2E 89 0E D9 03 seg000:0072 2E 88 36 DB 03 seq000:0077 2E 66 FF 0E D9 03 seg000:007D 9D seq000:007E 9C seg000:007F 2E FF 1E 4C 00 seq000:0084 OF 82 FC 01 seg000:0088 1E seq000:0089 06 seq000:008A 60 seg000:008B 9C seg000:008C 2E A0 E3 03 seg000:0090 3C 42 seq000:0092 75 1E

INT13 hooked

pushf

ah, 2

short ifRead

short ifRead

ah, 42h ; 'B'

far ptr loc_0

cs:3E3h, ah

cs:3E1h, al

cs:3D9h, cx

cs:3DBh, dh

10cret 284

al, cs:3E3h
al, 42h ; 'B'

short notExtendedRead

ds

es

dword ptr cs:3DDh, 0

dword ptr cs:3D9h, 0

dword ptr cs:3D9h

стр

jz

стр

jz

popf

jmp

MOV

MOV

MOV

MOV

mov

mov

dec

popf

pushf

call

push

push

pusha

pushf

mov

стр

jnz

jb

; DATA XREF: se

; Extended Read

; save function

; seq000:00481j

; save head

dword ptr cs:jmpOriq13h+1 ; jmp to th

; Read Sectors rom Drive

; DATA XREF: se 000:00101w ; seg000:001F1r ...

; CODE XREF: seg000:00431j

; save function number

; save track and sector

; jmp to the or ginal int 13h

; save number of sectors to read

; save function number

000:0023To

number

Sectors From Drive

BCD modification

5eg000:021D -og000:021D	CmnOndDatchDCD		
-00888-821D	contraction accurate	÷	; CODE XREF: seg000:00C5Tj
Sedaga 15			: sea000:00D6Ti
seg000:021D 0F B6 0E E1 03		MOVZX	CX, yee per ustatin
seg000:0222 C1 E1 07		shl	
seg000:0225			
seg000:0225	loc_225:		; CODE XREF: seg000:027Ejj
seg000:0225 26 66 81 3F 31 36 30 30	_	cmp	dwood ptr es:[bx], '0061' ; compare with BcdLibraryBoolean_EmsEnabled
seg000:022D 75 1C		jnz	shor: loc_24B ; finds hive's "1f"
seg000:022F 26 66 81 7F 04 30 30 32 30		спр	dwo ptr es:[bx+4], '0200'
seg000:0238 75 11		jnz	short loc_24B ; finds hive's "lf"
seq000:023A 26 66 C7 07 32 36 30 30		mov	dwood ptr es:[bx], '0062' ; patch to Bcd0SLoaderBoolean WinPEMode
seg000:0242 26 66 C7 47 04 30 30 32 32		mov	dwood ptr es:[bx+4], '2200'
seq 000: 024B			
5eq 000:024B	loc 24B:		; CODE XREF: seq000:022D1j
5eq 000: 024B	-		; sea000:0238↑j
seq000:024B 26 66 81 3F 6C 66 01 00		стр	dword ptr es:[bx], 1666Ch ; finds hive's "lf"
seq000:0253 75 14		jnz	shore roc zoz , search for /minim
5eq000:0255 26 66 81 7F 08 31 36 30 30		CMD	dword ptr es:[bx+8], '0061'
5ea000:025E 75 09		inz	short loc 269 : search for /MININT
5eq000:0260 26 66 C7 47 08 32 36 30 30		mov	dword ptr es:[bx+8], '0062'; patch key crc
5eq000:0269			
5ea000:0269	loc 269:		: CODE XREF: sea000:02531i
5eq000:0269			: seq000:025E1i
5eg000:0269 26 66 81 3F 2F 4D 49 4E		CMD	dword ptr es:[bx]. 'NIM/' : search for /MININT
5eq000:0271 75 08		inz	short loc 278
5eq000:0273 26 66 C7 07 49 4E 2F 4D		mov	dword ptr es:[bx]. 'M/NI' : patch it
5egAAA: 027B			
5egA00:0278	1oc 278:		: CODE XREE: segnage:02711i
5en000-0278 83 C3 04	100 1101	hhe	hy h
5en000:027F F2 A5		1000	Inc 225 : compare with BcdlibraruBoolean Ems

