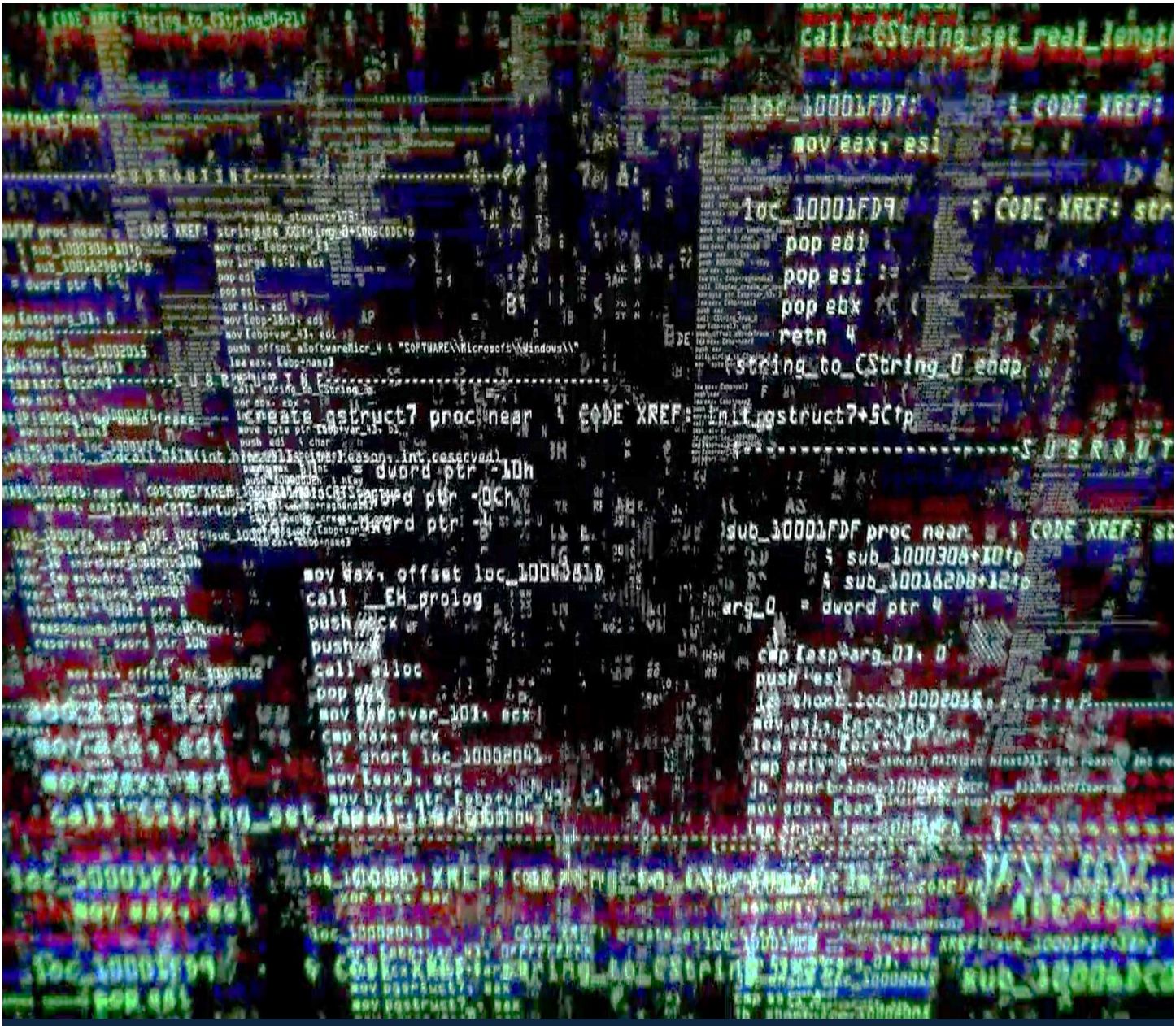

Windows Rootkits Guide

An attempt to summarize information about Windows rootkits

Artem Baranov
https://www.linkedin.com/in/aibaranov/

1/1/24

v1.0

Just an attempt to summarize information about known Windows Ring 0 rootkits using public information and my own knowledge. Unfortunately, a lot of interesting articles are no longer reached due to their age and web links leading to them are broken. It's worth adding that there are a higher number of Windows rootkits families in-the-wild, but information about them hasn't been published. Nevertheless, the table below covers the most notable instances, which were the first employing power Windows kernel tricks. Thanks to the guys who mirrored the articles and PoCs from the closed rootkit.com forum on GitHub.

Things that are out of scope of this document:

- Bootkits (MBR/UEFI/VBR/IPL) and their drivers; you can check out this blog post called Windows Bootkits Guide or other sources, for example, An In-Depth Look at Windows Kernel Threats by Trend;
- SMM (Ring -2), hypervisor (Ring -1), and hw/fw related rootkit concepts (Blue Pill, SMM rootkit); [25]
- Ring 3 (user mode) rootkits;
- BYOVD drivers and techniques of abusing legitimate Windows drivers;
- Drivers intended to help defeat Driver Signature Enforcement (DSE) or PatchGuard;
- Other malicious drivers that don't perform any modifications on the Windows kernel and its environment, for example, simple disk wipers, code injectors (with some exceptions), AV/EDR killers, backdoors allowing access to km address space, WFP and FS minifilter drivers that are only used to implement a private device stack w/o any OS modifications, etc (for example, Flame, Moriya, LuckyMouse, Exforel, ProjectSauron Remsec…); rootkits that use those techniques for self-defense are included.

Nevertheless, there are also some exceptions for earlier Windows user mode rootkits performing kernel mode manipulations (DKOM) via \Device\PhysicalMemory. Also, rootkits that employ the concept of Virtual/Hidden/Encrypted File System are included [24].

The following web resources made this document possible.

- Malpedia | https://malpedia.caad.fkie.fraunhofer.de/
- MITRE ATT&CK® | https://attack.mitre.org/
- KernelModeInfo forum | https://www.kernelmode.info/forum/
- rootkit_com site mirror | https://github.com/claudiouzelac/rootkit.com/tree/master/
- Virus Bulletin | https://www.virusbulletin.com/virusbulletin/
- An In-Depth Look at Windows Kernel Threats by Trend | https://documents.trendmicro.com/assets/white_papers/wp-an-in-depth-look-at-windows-kernel-threats.pdf

In addition, these resources are very useful for finding information about the aforementioned low-level malware.

A complete list of sources is provided in a separate section below. Some articles are given without web links, since they are no longer accessible due to their age, but I have my own repository with old rootkit articles, and it is available for download from my site https://www.artemonsecurity.com/useful.html.

Necessary reading skills: Windows Internals, malware analysis, reverse engineering at medium+ level.

Contacts:

Artem Baranov, an independent security researcher

artembaranovex@gmail.com

https://www.linkedin.com/in/aibaranov/

Contents

Rootkit techniques 3
The Table with Rootkit Families 6
References 13

«Adversaries may use rootkits to hide the presence of programs, files, network connections, services, drivers, and other system components.» MITRE ATT&CK T1014

Rootkit techniques

[T1] Intercepting system services to control calls of basic Windows kernel functions (system services [3])

- [T1.a] Modifying SSDT (*KiServiceTable*) [1] [2]
- [T1.b] MSR_SYSENTER (IA32_SYSENTER_EIP, CS) for *sysenter* on x86 [1] [2] [16]
- [T1.c] KTHREAD.ServiceTable [4]
- [T1.d] IDT[0x2E] system service interrupt [1] [2] [4]
- [T1.e] Inline patching of *KiSystemService* or *KiFastCallEntry* (x86) [1] [2]
- [T1.f] *Nt** functions from SSDT

[T2] Direct Kernel Object Manipulation (DKOM) to manipulate Windows kernel structures [20]

- [T2.a] Unlinking drivers from *PsLoadedModulesList* (LDR_DATA_TABLE_ENTRY) [1] [2]
- [T2.b] processes from *PsActiveProcessHead* [1] [2]
- [T2.c]* threads from *KiWaitInListHead, KiWaitOutListHead, KiDispatcherReadyListHead* [1] [2] [6] [7]
- [T2.d] Modifying access token [1] [2]
- [T2.e] Removing objects from *Ob* object directory
- [T2.f] driver objects from the list of driver objects
- [T2.g] device objects from the list of device objects
- [T2.h] Forging ETHREAD fields [1] [2]
- [T2.i] EPROCESS fields [1] [2]
- [T2.j] Erasing items in *PspCidTable* [2]
- [T2.k] handles in the process handle table
- [T2.l] Intercepting object type dispatch functions (procedures) [13]
- [T2.m] Forging DRIVER_OBJECT fields
- [T2.n] Hijacking driver object [13]
- [T2.o] Hijacking device object [13]

[T3] Inline patching kernel mode code (run-time patching, inline hooking, splicing) [1] [2]

- [T3.a] Ntoskrnl - * *Nt**, *IofCallDriver*, *IofCompleteRequest*, etc. [7]
- [T3.b] FSD – Ntfs.sys, Fastfat.sys and attached filter, minifilter drivers (Filter Manager) [1] [2] [5]
- [T3.c] TCP/IP, NDIS - Tcpip.sys, Ndis.sys [1] [2] [4]
- [T3.d] IP Filter Driver - Ipfilterdriver.sys

[T4] Intercepting driver object major functions

- [T4.a] FSD – Fastfat.sys, Ntfs.sys to hide files [1] [2] [5]
- [T4.b] TDI Tcpip.sys, Ndis.sys, also NDIS_OPEN_BLOCK and NDIS_PROTOCOL_BLOCK handlers to manipulate network communications [2] [17] [18]

[T4.c] Disk port drivers - atapi.sys, ataport.sys, storport.sys, scsiport.sys to hide disk sectors [5]

[T4.d] Fast I/O Dispatch Routine (FastIoDeviceControl) *AfdFastIoDeviceControl* of Afd.sys to manipulate network traffic

[T4.e] Network Store Interface (NSI) driver nsiproxy.sys to hide TCP ports [22]

[T4.f] Null.sys to hide rootkit activity

[T4.g] LiveKd debugger driver

[T4.h] FS Filter Manager fltmgr.sys [4]

[T5] Intercepting IDT/ISR [1] [2] [5]

[T6] Setting up itself as a filter driver for (attaching to the corresponding device stack for self-defense) [1]

[T6.a] File System Driver (FSD), legacy or minifilter [2]

[T6.b] Volume Manager (volmgr.sys, volmgrx.sys) [4]

[T6.c] TCP/IP stack, NDIS (tcpip.sys, ndis.sys) [4] [17]

[T6.c] NSI driver nsiproxy.sys

[T7] Using Windows kernel callbacks (in case of blocking access to system resources and self-protection) [4] [9]

[T7.a] CmRegistry, LoadImageNotify, ObRegisterCallbacks

[T8] Using and hiding NTFS Alternate Data Streams (ADS) [4]

[T9] Keylogger (attaching to \Device\KeyboardClass0) [1] [2]

[T10] Windows IP Filtering [1]

[T11] Disabling Windows kernel callbacks (create process, create thread, load image, registry ops, kernel object, object manager) to blind AV and EDR products [9]

[T11.a] LoadImageNotify, CreateThreadNotify, CreateProcessNotify, CmRegistry, *ObRegisterCallback*

[T12] Other tricks

[T12.a] *ObMakeTemporaryObject* to remove the driver object's name from the \Driver\ object manager directory

[T12.b] Disabling WFP callout drivers (via *netio!gwfpGlobal*) [10]

[T12.c] Disabling Event Tracing (ETW) (via *nt!EtwpHostSiloState*) [11]

[T12.d] Disabling System Loggers (via *nt!EtwpActiveSystemLoggers*)

[T12.e] Disabling FS minifilter drivers (via unlinking the appropriate structures) [12]

[T12.f] Disabling Image Verification Callbacks

[T12.g] Hidden File System (VFS) [5] [24]

[T12.h] Hiding services by unlinking the corresponding SERVICE_RECORD structure in the context of the Services.exe process [14]

[T12.i] Preventing writing kernel memory dumps by registering its callback with *KeRegisterBugCheckReasonCallback*

[T12.j] Replacing HHIVE.GetCellRoutine pointer to get control over system registry operations [19]

[12.k] Disabling FS minifilter drivers via *FltUnregisterFilter*

*In fact, this technique is very rare, because if a thread is removed from one of those lists, it also becomes hidden to the thread scheduler and won't get any CPU time. The only rootkit using it is 90210's phide2 that runs its own thread scheduler after copying the necessary ntoskrnl functions (utilizing pullout engine) to a separate buffer to serve the hidden thread. In spite of this, it can be useful for anti-rootkits (for example, klister) that go through those lists, exposing hidden processes (unlinked from *PsLoadedModulesList*).

The Table with Rootkit Families

Has known authors

Utilize \Device\PhysicalMemory

Belong to the same malware family or TA (attribution according to the public information): FU, Phide, Rustock, Tdss (TDL, Alureon), BlackEnergy, Attributed to the NSA

Name	Year	Platform	Features	Author/Detection	Additional details
NTRootkit [26] Originally concept	1999	x86	[T1.a] [T1.d] [T9]	Greg Hoglund	Do not confuse this original rootkit concept and x86 SSDT hooking rootkits based on this widely known technique with the NtRootkit malware family of some AV companies that use it as a generic detection name for the rootkits that they have not classified.
FU [27] [28]	2004	x86	[T2.a] [T2.d]	fuzen_op --- WinNT/FURootkit.A	
Phide [29] Concept	2004	x86	[T2.b] [T2.h]	90210	Kernel memory access via \Device\PhysicalMemory
Phide2 [30] Concept	2004	x86	[T2.b] [T2.c]	90210	Loads another copy of ntoskrnl and runs its own thread scheduler [6]
FUTo [31]	2005	x86	[T2.a] [T2.d] [T2.j] [T2.k]	Peter Silberman C.H.A.O.S. --- Win32.Fuzen.a	
Myfip [32]	2005	x86	[T2.a]	W32/Myfip	Kernel memory access via \Device\PhysicalMemory
Fanbot [32]	2005	x86	[T2.a]	W32/Fanbot	Kernel memory access via \Device\PhysicalMemory --- There are many more rootkits utilizing this technique
Shadow Walker [33] [34] [35] Concept	2005	x86	[T5]	Greg Hoglund James Butler	Intercepts access to Translation Lookaside Buffer (TLB)
Sony XCP rootkit [36] [37] [38] [39]	2005	x86	[T1.a]	SecurityRisk.First4DRM	

Apropos [40]	2005	x86	[T3.a]	?	
Rustock.A [41]	2006	x86	[T1.b] [T1.c] [T2.a] [T3] [T4.a] [T8]	Backdoor.Rustock.A	Polymorphic packer Malpedia
Rustock.B [42] [43] [44]	2006	x86	[T1.b] [T1.c] [T2.a] [T3] [T4.a] [T4.b] [T8]	Backdoor.Rustock.B	Polymorphic packer Malpedia
phide_ex [45] [46] Concept	2006	x86	[T1] [T2] [T4.a]	PE386/MS-REM --- WinNT/Rustock.C	
RKDemo Concept	2006	x86	?	MP-ART EP_X0FF	
Unreal.A [47] Concept	2007	x86	[T2.a] [T2.e] [T2.f] [T2.g] [T2.h] [T6.a] [T8]	MP-ART EP_X0FF --- DSSDetection Hacktool.Unreal.A	
Rustock.C originally named by mistake, doesn't belong to the Rustock family [48] [49] [50] [51]	2007	x86	[T1.e] [T2.l] [T3.b] [T3.c]	PE386/MS-REM --- WinNT/Rustock.D Win32.Ntldrbot Win32.Rustock.a	Heavy code obfuscation+ multiple encryption layers + hardware locking + anti-debugging tricks + anti-patching tricks Malpedia
Haxdoor [52] [53] [54] [55] [56] [57]	2007	x86	[T1.a] [T2.a]	WinNT/Haxdoor	
BlackEnergy [57] [58] [59]	2007	x86	[T1.1]	?	S0089 Malpedia
Srizbi [60]	2007	x86	[T1.f] [T4.a] [T4.b]	Trojan.Srizbi	The first itw malware that operates fully from Ring 0 + runs its own copy of ndis.sys + Features its own private TCP/IP stack
Alman [61] [62]	2007	x86	[T1.a]	Win32/Alman Virus:Win32/Almanahe Virus.Win32.Alman	

TDL1 **Tdss** **Tidserv** **Alureon** [63]	2008	x86	[T1.f] [T2.a] [T6.b] [T6.c]	Rootkit.Win32.Clbd.a	Malpedia
TDL2 **Tdss** **Tidserv** **Alureon** [63] [64] [65]	2009	x86	[T1.f] [T2.a] [T3.a] [T6.b] [T6.c]	Win32/Alureon.BK Backdoor.Tidserv	Malpedia
TDL3 **Tdss** **Tidserv** **Alureon** [63] [66] [67] [68] [69]	2009	x86	[T4.c] early versions [T2.n] [T2.o] [T12.g]	Trojan:Win32/Alureon.CT Backdoor.Tidserv Win32/Olmarik.SC	Infects disk port drivers (atapi.sys, iastor.sys) Malpedia
ZeroAccess **ZAccess** **Sirefef** shares similarities with TDL3 [70] [71] [72] [73] [74]	2009	x86	[T2.m] [T2.n] [T2.o] [T12.g]	Trojan:Win32/Sirefef Virus.Win32.ZAccess BackDoor.Maxplus	Early versions replaced system drivers + infects system drivers in TDL3 manner + creates \Device__max++ TDL3 removing routine + creates hidden encrypted volume S0027 Malpedia
Papras **Gozi** **Ursnif** [57] [75]	2009	x86	[T1.a]	Win32/Ursnif.CU PSW.Papras.AO	S0386 Malpedia
Koutodoor [76]	2009	x86	[T2.l]	Win32/Koutodoor	Polymorphic code + After each reboot the rootkit moves the driver to another file
Stuxnet [77] [78] attributed to the NSA	2010	x86	[T6.a]	Trojan:WinNT/Stuxnet Rootkit.Win32.Stuxnet	Signed by a stolen cert S0603 Malpedia
Bubnix [79]	2010	x86	[T2.l] [T4.a]	Win32/Bubnix WinNT/Bubnix	
Alureon.DO [80] TDL2 clone	2010	x86	[T1.f] [T2.a] [T3]	Win32/Alureon.DO Win32.TDSS.bwkw Win32/Olmarik.TL	Malpedia
Cutwail [81] [82] [83]	2010	x86	[T1.a] [T4.a]	WinNT/Cutwail	Malpedia
Sality [84]	2010	x86	[T10]	Trojan:WinNT/Sality Virus.Win32.Sality	Blocks network packets belonging to AV vendors Malpedia
BlackEnergy v2 [85]	2010	x86	[T1.c] [T3.a]	Backdoor:WinNT/Phdet.A RTKT_RUSTOCK.SMB BackDoor.BlackEnergy	S0089 Malpedia
Ramnit [86] [87] [88]	2010	x86	[T1.a]	Worm:Win32/Ramnit.A Win32/Ramnit.A	Acts as AV killer + Restores SSDT pointers to disable AV hooks Malpedia

Necurs [89] [90] [91] [92] [93] [94]	2011	x86 x64	[T1.a] x86 [T6.a] [T7.a] x64	Trojan:WinNT/Necurs Rootkit.Win32.Necurs	Creates \Device\NtSecureSys Starts with the highest priority as Boot Bus Extender Malpedia
Duqu [95] [96] attributed to the NSA	2011	x86	Tricky code injection	Trojan:WinNT/Duqu Trojan.Win32.Duqu	Files signed by a stolen cert + unique file names S0038 Malpedia
Festi [5] [97] [98]	2012	x86	[T1.a] [T6.a]	Win32/Rootki.Festi Rootkit.Win32.Tent	
WindowsRegistryRootkit Concept [99] [100]	2012	x86	[T3.c]	Cr4sh --- Trojan:Win32/Leivion.I Trojan.Win32.Diple.fzxp	Stored in the Windows registry + Infects Windows drivers in memory
Avatar [101] [102] [103]	2013	x86	[T2.m] [T4.c] [T12.g]	Rootkit.Win32.Avatar BackDoor.Avatar Win32/Rootkit.Avatar	Capable of being loaded as fileless rootkit + Infects other system drivers + Employs VM detection via *MmMapIoSpace* + Has its own SDK + Relocates disk port driver + Infects a disk port driver in memory
Sednit [104] [105] SVR-aligned TA	2014	x86 x64	[T1.a] [T6.a]	Rootkit.Win32.Agent.ekik Win32/Rootkit.Agent	G0007 Malpedia
considered one of the most sophisticated rotkits **Turla (Snake) Uroburos** linked to Agent.BTZ state-sponsored Russia-aligned [106] [107] [108] [109] [110] [111]	2014	x86 x64	[T1.f] [T3.a] [T4.b] [T12.g]	Rootkit.Win64.Turla Backdoor:WinNT/Turla Win64/Turla Rootkit:Uroburos	Creates its own (clean) copy of *KiServiceTable* + Bypasses PatchGuard by hooking *KeBugCheckEx* + Bypasses DSE by abusing Oracle VirtualBox driver VBoxDrv.sys + Relocates itself into System process + Uses the interrupt 0xC3 for the hooking engine + Incorporates Udsi86 disasm G0010 Malpedia
Duqu 2.0 [112] attributed to the NSA	2015	x86 x64	[T6.c]		termport.sys S0038 Malpedia
Grayfish [113] [114] Equation Group **TA**	2015	x86 x64	[T2.m] [T12.a]	Trojan.Win32.GrayFish	\Driver\msvss
EquationDrug state-sponsored NSA-aligned [115] [116]	2015	x86 x64	[T6.c]	Trojan:WinNT/Eqtonex.A Win32/Equdrug.A Trojan.Equdrug	mstcp32.sys \Device\Mstcp32 Malpedia
Hacker's Door [117]	2017	x86 x64	[T3.d]	Win64/Hackdoor.A Backdoor:Win64/Hackdoor.A	kifesEn.sys + Instead of intercepting functions and callbacks of Ndis, Tcpip drivers, patches IpFltDrv.sys

Derusbi [118] [119] [120] state-sponsored	2017	x86 x64	[T1.a] x86 [T4.a] [T4.b] [T4.e]		Files signed by stolen certs+ Anti-debug tricks S0021 Malpedia
Wingbird [121] [122] state-sponsored	2017	x86	[T1.a] to remove hooks	Backdoor:Win32/Wingbird	The code is heavy obfuscated + Self-modifying (mutable) code + Removes AVers SSDT hooks + Covert code injection S0176
Finfisher state-sponsored [122] [123] [124] [125]	2017	x86	Tricky code injection	Backdoor:Win32/Finfish Backdoor.Win32.FinFish Win32/FinSpy	The code is heavy obfuscated + Self-modifying (mutable) code S0182 Malpedia
Purple Fox [126]	2017	x64	[T12.k]	Trojan.Win64.PURPLEFOX.YACAM PUA:Win32/Creprote	
Cahnadr [127] state-sponsored Slingshot APT	2018	x86 x64	[T1.c] x86 [T3.c] [T4.f] x64 [T4.g] [T12.i]	Backdoor.Win32.Slingshot Win64/Slingshot Trojan.Slingshot	Loaded covertly through MSR_LSTAR handler *syscall* [16][17] Abuses 3rd party drivers to bypass DSE + Checks Ntoskrnl and Win32k for hooks+ Patches Null.sys in memory + Receives commands through its CmRegisterCallback (x86) + Anti-debug tricks
HideProc [128] [129]	2018	x86	[T2.b]	Win32/HideProc	
CEIDPageLock [130]	2018	x86	[T4.d]	Backdoor.Graybird Trojan.NtRootKit.19661 PUA:Win32/Kuping	Packed with VMProtect Signed by stolen cert
Zacinlo [131]	2018	x86 x64	[T6.a] [T7.a]	Troj/Zacinlo-A	radardt.sys \Device\DrvProtect Signed Malpedia
Autochk attributed to Emissary Panda [132]	2019	x64	[T4.e] [T4.h]	Rootkit.Win64.ZXShell.e Trojan:Win32/Zxshell Win64/ZxShell.AH	Signed + Redirects requests to hidden files to legitimate ones
FK_Undead [133] [134]	2020	x64	[T7.a]	Rootkit.Win32.Agent.elyj	Packed with VMProtect
Cronos [135]	2021	x64	[T2.a] [T2.b] [T2.d]	Win64/Rootkit.Cronos Win64:CronosRootkit-A [Rtk]	

Demodex [136]	2021	x86 x64	[T4.e] [T3] Pci.sys [T6.a] [T7.a] [T12.h]		Loaded via abusing Cheat Engine driver dbk64.sys
DirtyMoe [137]	2021	x86 x64	[T2.a] [T4.a] [T6.a] [T12.j]	Win32:DirtyMoe-C Win64/Rootkit.Agent.T Rootkit.Win64.Agent.bfo	The driver is written on disk every time the system starts and is deleted once it's loaded + Executables are signed by stolen certs Malpedia
FiveSys [138]	2021	x64	[T7.a]	Trojan.Win64.PACSYS.A Rootkit.Agent.AJIQ	Signed S0618
FudModule attributed to Lazarus [139] [140] [141] [142] [143]	2022	x86 x64	[T2.h] [T3.b] [T11.a] [T12.b] [T12.c] [T12.d] [T12.e] [T12.f]	Acts from um abusing 3rd party drivers that enable it to operate on Ring 0 memory utilizing physical<->virtual address translation --- Rootkit/Win.Agent.C5192169 Rootkit/Win.Agent.C5177679	Operates via BYOVD (ene.sys, appid.sys) + Uses physical memory to access km address space + Modifies KTHREAD.PreviousMode to allow um code accessing km address space via *ZwWriteVirtualMemory* Removes AVers callbacks + Disables ETW + Unlinks FS minifilter drivers Malpedia
Fire Chili [144] attributed to Deep Panda	2022	x86 x64	[T2.b] [T6.a] [T6.c] [T7.a]	Rootkit.Win64.Agent.bjm Hacktool.Rootkit	Drivers are signed by stolen certs + Loaded via BYOVD Malpedia
InvisiMole [145]	2022	x64	[T6.a] [T11.a] [T12.b] [T12.h]	Win64/InvisiMole	S0260 Malpedia
Daxin [146]	2022	x86 x64	[T4.b]	Trojan.Win64.Agentb.bwb Backdoor.Daxin	Features its own private TCP/IP stack Malpedia
WINNKIT [147] [148] state-sponsored, Winnti TA	2022	x86 x64	[T2.a] [T2.f] [T2.g] [T4.b] [T4.f]	Backdoor.Multi.Winnti Win64:Winnti-L [Trj]	Signed + Also implements its own private TCP/IP stack Malpedia


1999 NtRootkit
2004 FU Phide Phide2
2005 FUTo Myfip Fanbot Shadow Walker Sony XCP Apropos
2006 Rustock.A Rustock.B phide_ex RKDemo
2007 Unreal.A Rustock.C Haxdoor BlackEnergy Sony XCP Srizbi Alman
2008 TDL1
2009 TDL2 TDL3 ZeroAccess Papras Koutodoor
2010 Stuxnet Bubnix Alureon.DO Cutwail Sality BlackEnergy 2 Ramnit
2011 Necurs Duqu
2012 Festi WindowsRegistry
2013 Avatar
2014 Sednit Turla
2015 Duqu 2 Grayfis EquationDrug
2017 Hacker's Door Derusbi Wingbird Finfisher Purple Fox
2018 Cahnadr HideProc CEIDPageLock Zacinlo
2019 Autochk
2020 FK_Undead
2021 Cronos Demodex DirtyMoe FiveSys
2022 FudModule Fire Chili InvisiMol Daxin WINNKIT

References

[1] Greg Hoglund, James Butler | *Rootkits: Subverting the Windows Kernel*

[2] Bill Blunden | *The Rootkit Arsenal*

[3] Gary Nebbett | *Windows NT/2000 Native API Reference*

[4] David A. Solomon, Mark Russinovich, Alex Ionescu, Pavel Yosifovich, Andrea Allievi | *Windows Internals (7th edition)*

[5] Alex Matrosov, Eugene Rodionov, Sergey Bratus | *Rootkits and Bootkits: Reversing Modern Malware and Next Generation Threats*

[6] Alexander Tereshkin (90210) | Bypassing Klister 0.4 With No Hooks or Running a Controlled Thread Scheduler

[7] Microsoft | Windows Research Kernel (WRK)

[8] Ron Ben Yizhak of Deepinstinct | #NoFilter - Abusing Windows Filtering Platform for Privilege Escalation

[9] Jonathan Johnson | Understanding Telemetry: Kernel Callbacks

[10] Microsoft | Windows Filtering Platform

[11] Microsoft | Event Tracing for Windows (ETW)

[12] Microsoft | Filter Manager Concepts

[13] Sherif Magdy, Mahmoud Zohdy of Trend Micro | An In-Depth Look at Windows Kernel Threats

[14] Eugene Wineblat | Service Hiding

[15] Michal Poslušný of ESET | Signed kernel drivers – Unguarded gateway to Windows' core

[16] Rotem Salinas of Cyberark | Fantastic Rootkits: And Where to Find Them (Part 1)

[17] 90210 | Rootkits: Attacking Personal Firewalls

[18] F-Secure | Spam from the kernel

[19] Ph4nt0m Security Team | https://pstgroup.blogspot.com/2007/07/tips.html

[20] James Butler of HBGary | DKOM (Direct Kernel Object Manipulation)

[21] Mariusz Burdach, SANS Institute | FU rootkit

[22] CardMagic(Edward) | PortHidDemo_Vista

[23] Kimmo Kasslin of F-Secure, Elia Florio of Symantec | When malware meets rootkits

[24] MITRE ATT&CK | Hide Artifacts: Hidden File System

[25] Alexander Tereshkin and Rafal Wojtczuk | Introducing Ring -3 Rootkits

✨ NTRootkit

[26] GitHub | Rootkit_com mirror hoglund

✨ FU

[27] Mariusz Burdach, SANS Institute | FU rootkit

[28] GitHub | Rootkit_com mirror Fuzen_op

✨ Phide

[29] GitHub | Rootkit_com mirror 90210 |

✨ Phide2

[30] GitHub | Rootkit_com mirror 90210

✨ FUTo

[31] GitHub | Rootkit_com mirror petersilberman

✨ Myfip

[32] F-Secure | Myfip.H

✨ Fanbot

[32]

✨ Shadow Walker

[33] Sherri Sparks, Jamie Butler | «SHADOW WALKER» Raising The Bar For Rootkit Detection

[34] GitHub | Rootkit_com mirror hoglund

[35] 90210 | Defeating Shadow Walker

✨ Sony XCP rootkit

[36] Microsoft | Sony, Rootkits and Digital Rights Management Gone Too Far

[37] Symantec | SecurityRisk.First4DRM

[38] Mark Russinovich | Inside Sony's rootkit

[39] Deirdre K. Mulligan, Aaron Perzanowski of University of Michigan | The Magnificence of the Disaster: Reconstructing the Sony BMG

✨ Apropos

[40] F-Secure | Apropos

✨ Rustock.A

[41] Elia Florio, Prashant Pathak of Symantec | Raising the bar: Rustock and advances in rootkits

✨ Rustock.B

[42] Clifford Weaver | Finishing up Rustock.B

[43] Ken Chiang, Levi Lloyd of Sandia National Laboratories | A Case Study of the Rustock Rootkit and Spam Bot

[44] Frank Boldewin | A Journey to the Center of the Rustock.B Rootkit

✨ phide_ex

[45] Yan Wen, Jinjing Zhao, Huaimin Wang | Implicit Detection of Hidden Processes with a Local-Booted Virtual Machine

[46] Carl Jongsma of Sûnnet Beskerming | Undetectable Rootkit

✨ Unreal.A

[47] Carl Jongsma of Sûnnet Beskerming | Undetectable Rootkit

✨ Rustock.C

[48] Vyacheslav Rusakoff of Dr.Web | Win32.Ntldrbot (aka Rustock.C)

[49] Lukasz Kwiatek, Stanislaw Litawa of ESET | 'Yet another Rustock analysis ...'

[50] Chandra Prakash of Sunbelt Software | Your filters are bypassed: Rustock.C in the kernel

[51] Chandra Prakash of Sunbelt Software | Kernel mechanics of Rustock

✨ Haxdoor

[52] Microsoft | WinNT/Haxdoor

[53] Oleg Zaitsev | Backdoor.Win32.Haxdoor.gu

[54] Desmond Lobo, Paul Watters and Xin-Wen Wu | Identifying Rootkit Infections Using Data Mining

[55] Ken Dunham of iSIGHT Partners | Introduction to advanced memory analysis

[56] Ken Dunham of iSIGHT Partners | Memory analysis - examples

[57] Sami Al-Shaheri, Dale Lindskog, Pavol Zavarsky of Concordia University College of Alberta | A Forensic Study of the Effectiveness of Selected Anti-Virus Products Against SSDT Hooking Rootkits

✨ BlackEnergy

[58] Jose Nazario of Arbor Networks | BlackEnergy DDoS Bot Analysis

[57]

[59] Secureworks | BlackEnergy Version 2 Threat Analysis

✨Srizbi

[60] Kimmo Kasslin of F-Secure, Elia Florio of Symantec | Spam from the kernel

✨Alman

[61] kernelmode_info | Virus.Win32.Alman.b

[62] Desmond Lobo, Paul Watters and Xin-Wen Wu | Identifying Rootkit Infections Using Data Mining

✨TDL1

[63] Vyacheslav Rusakov, Sergey Golovanov of Kaspersky | TDSS

✨TDL2

[64] Lavasoft | An Analysis of Rootkit Technologies: Part 3

[63]

[65] kernelmode_info | Rootkit TDL 2 (Alias TDSS, Alureon.BK)

✨TDL3

[66] kernelmode_info | Rootkit TDL 3 (alias TDSS, Alureon.CT, Olmarik)

[67] Dr.Web | BackDoor.Tdss.565 and its modifications (aka TDL3)

[63]

[68] Ace Portuguez of F-Secure | The Case of Trojan DownLoader TDL3

[69] Nguyễn Phố Sơn (t4l) | A Detailed Analysis of TDL Rootkit 3rd Generation

✨ZeroAccess

[70] kernelmode_info | Rootkit ZeroAccess (alias MaxPlus, Sirefef)

[71] Marco Giuliani of PrevX | ZeroAccess – an advanced kernel mode rootkit

[72] Marco Giuliani of Webroot | TDL3 and ZeroAccess: More of the Same?

[73] Vasily Berdnikov (vaber) | MAX++ sets its sights on x64 platforms

[74] Sean Hittel and Rong Zhou of Symantec | Trojan.ZeroAccess Infection Analysis

✨ Papras

[57]

[75] Desmond Lobo, Paul Watters and Xin-Wen Wu | Identifying Rootkit Infections Using Data Mining

✨ Koutodoor

[76] Aditya Kapoor, Rachit Mathur of McAfee | Predicting the future of stealth attacks

✨ Stuxnet

[77] Artem Baranov | Stuxnet drivers: detailed analysis

[78] Alexander Gostev, Igor Kuznetsov | Stuxnet/Duqu: The Evolution of Drivers

✨ Bubnix

[79] kernelmode_info | Nixoa/Bubnix Rootkit

✨ 4DW4R3 (TDL 2 clone)

[80] kernelmode_info | Rootkit 4DW4R3 (TDL 2 clone)

✨ Cutwail

[81] Kyle Yang of Fortinet | Cut the Cutwail

[82] Aditya Kapoor, Rachit Mathur of McAfee | Challenges in Kernel-Mode Memory Scanning

[83] kernelmode_info | Win32/Cutwail

✨ Sality

[84] Artem Baranov | Sality rootkit analysis

✨ BlackEnergy v2

[85] kernelmode_info | WinNT/BlackEnergy

✨ Ramnit

[86] kernelmode_info | Win32/Ramnit

[87] Chao Chen of Fortinet | Ramnit bot

[88] Symantec | W32.Ramnit analysis

✨ Necurs

[89] Artem Baranov | Necurs rootkit under microscope

[90] kernelmode_info | Necurs - another x64 rootkit

[91] Peter Ferrie of Microsoft | The curse of Necurs, part 1

[92] Peter Ferrie of Microsoft | The curse of Necurs, part 2

[93] Peter Ferrie of Microsoft | The curse of Necurs, part 3

[94] Vyacheslav Zakorzhevsky of Kaspersky | An unlikely couple: 64-bit rootkit and rogue AV for MacOS

✨ Duqu

[95] Boldizsár Bencsáth, Gábor Pék, Levente Buttyán, Márk Félegyházi of CrySyS | Duqu: A Stuxnet-like malware found in the wild

[96] Guillaume Bonfante, Jean-Yves Marion, Fabrice Sabatier, Aurélien Thierry | Analysis and Diversion of Duqu's Driver

✨ Festi

[5]

[97] Eugene Rodionov, Alexander Matrosov of ESET | King of Spam: Festi Botnet Analysis

[98] Check Point | Return of the Festi Rootkit

✨ WindowsRegistryRootkit

[99] Cr4sh | WindowsRegistryRootkit on GitHub

[100] Cr4sh | Applied anti-forensics: rootkits and kernel vulnerabilities

✨ Avatar

[101] Aleksandr Matrosov, Anton Cherepanov of ESET | Mysterious Avatar rootkit with API, SDK, and Yahoo Groups for C&C communication

[102] Aleksandr Matrosov, Anton Cherepanov of ESET | Avatar rootkit: the continuing saga

[103] kernelmode_info | Win32/Rootkit.Avatar

✨ Sednit

[104] Artem Baranov | A note about Sednit rootkit

[105] ESET | En Route with Sednit Part 3: A Mysterious Downloader

✨ Turla

[106] deresz, tecamac | Uroburos: the snake rootkit

[107] Adam Crosser of praetorian | Developing a Hidden Virtual File System Capability That Emulates the Uroburos Rootkit

[108] hfiref0x | TDL (Turla Driver Loader)

[109] G Data | Uroburos Highly complex espionage software with Russian roots

[110] G Data | Uroburos – Deeper travel into kernel protection mitigation

[111] Paul Rascagneres of G DATA | Uroburos

✨ Duqu 2.0

[112] Kaspersky | The Mystery of Duqu 2.0: a sophisticated cyberespionage actor returns

✨ Grayfish

[113] Artem Baranov | GrayFish rootkit analysis

[114] Check Point | A Deep Dive Into DoubleFeature, Equation Group's Post-Exploitation Dashboard

✨ EquationDrug

[115] Artem Baranov | EquationDrug rootkit analysis (mstcp32.sys)

[116] Kaspersky | Inside the EquationDrug Espionage Platform

✨ Hacker's Door

[117] BlackBerry | Threat Spotlight: Opening Hacker's Door

✨ Derusbi

[118] Airbus | Newcomers in the Derusbi family

[119] Joe Desimone, Gabriel Landau | Kernel Mode Threats And Practical Defences

[120] CrowdStrike | Deep Panda

✨ Wingbird

[121] Artem Baranov | Wingbird rootkit analysis

[122] Andrea Allievi and Elia Florio of Microsoft | FinFisher exposed: A researcher's tale of defeating traps, tricks, and complex virtual machines

✨ Finfisher

[122]

[123] Filip Kafka of ESET | ESET's guide to deobfuscating and devirtualizing FinFisher

[124] Microsoft | Office 365 Advanced Threat Protection defense for corporate networks against recent Office exploit attacks

[125] Kaspersky | FinSpy: unseen findings

✨ Purple Fox

[126] Trend | A Look Into Purple Fox's New Arrival Vector

✨ Cahnadr

[127] Kaspersky | The Slingshot APT

✨ HideProc

[128] dhruval | HideProc on GitHub

[129] Ignacio Sanmillan of ESET | Ramsay: A cyber-espionage toolkit tailored for air-gapped networks

✨ CEIDPageLock

[130] Israel Gubi of Check Point | CeidPageLock: A Chinese RootKit

✨ Zacinlo

[131] Claudiu Cobliș, Cristian Istrate, Cornel Punga, Andrei Ardelean of Bitdefender | Six Years and Counting: Inside the Complex Zacinlo Ad Fraud Operation

✨ Autochk

[132] Ori Damari (repnz) | Autochk Rootkit Analysis

✨ FK_Undead

[133] Lab52 | Recent FK_Undead rootkit samples found in the wild

[134] Security Leopard | Legendary private servers hide hidden dangers, and the undead virus is raging in the arena

✨ Cronos

[135] XaFF-XaFF | Cronos-Rootkit on GitHub

✨ Demodex

[136] Mark Lechtik, Aseel Kayal, Paul Rascagneres, Vasily Berdnikov of Kaspersky | GhostEmperor: From ProxyLogon to kernel mode

✨ DirtyMoe

[137] Martin Chlumecký of Avast | DirtyMoe: Rootkit Driver

✨ FiveSys

[138] Cristian Alexandru ISTRATE, Balazs BIRO, Rares Costin BLEOTU, Claudiu Stefan COBLIS of Bitdefender | Digitally-Signed Rootkits are Back – A Look at FiveSys and Companions

✨ FudModule

[139] AhnLab | Lazarus Group's Rootkit Attack Using BYOVD

[140] Jan Vojtěšek of Avast | Lazarus and the FudModule Rootkit: Beyond BYOVD with an Admin-to-Kernel Zero-Day

[141] Jonathan Johnson | Understanding Telemetry: Kernel Callbacks

[142] Pierre Ciholas, Jose Miguel Such, Angelos K. Marnerides, Benjamin Green, Jiajie Zhang, Utz Roedig | Fast and Furious: Outrunning Windows Kernel Notification Routines from User-Mode

[143] Dylan (@batsec) of MDSec | Bypassing Image Load Kernel Callbacks

✨ Fire Chili

[144] Rotem Sde-Or and Eliran Voronovitch of Fortinet | New Milestones for Deep Panda: Log4Shell and Digitally Signed Fire Chili Rootkits

✨ InvisiMole

[145] Michal Poslušný | Signed kernel drivers – Unguarded gateway to Windows' core

✨ Daxin

[146] Symantec | Daxin Backdoor: In-Depth Analysis, Part One

✨ Winnti

[147] Stéfan Le Berre of ExaTrack | From tweet to rootkit

[148] Chen Erlich, Fusao Tanida, Ofir Ozer, Akihiro Tomita, Niv Yona, Daniel Frank, Assaf Dahan of Cybereason | Operation CuckooBees: A Winnti Malware Arsenal Deep-Dive