



Windows Rootkits

Chris Ries
Security Research Engineer
VigilantMinds Inc.

Introduction

- **What I do**
 - Research new vulnerabilities, malware, and other security threats
 - Create defensive measures
 - Evaluate security software
- **What this talk is about**
 - Windows rootkits
 - How they are used
 - How they work
 - Defensive measures

What is a Rootkit?

- **“A rootkit is a set of programs and code that allows a permanent or consistent, undetectable presence on a computer”**
- **Goals:**
 - **Hide malicious resources**
 - **Processes, files, registry keys, open ports, etc.**
 - **Provide hidden backdoor access**

Brief History

- **Early rootkits targeted UNIX OSes**
 - “Kits” to attain and maintain “root” access to machines
 - Replaced login, ls, ps, netstat, etc. to give an attacker hidden access
 - Eventually moved towards kernel
- **Windows popularity brought Windows rootkits**

Why So Popular?

- **Worms, trojans, malware are utilizing rootkits**
 - Presence becomes hidden
 - Machines stay infected longer -> can send spam and steal info longer -> more money for attacker
- **Some commercial software adopts rootkit technology**
 - Sony DRM software



How Rootkits Are Used

Stages of An Attack

- 1. Vulnerability in a system is discovered**
- 2. Vulnerability is exploited to gain access to the system**
- 3. Attacker gains a foothold on the system by escalating privileges, installing backdoor, etc.**
- 4. Attacker utilizes system access to steal information, launch other attacks, etc.**
- 5. Compromise is discovered, and incident response is executed**

Where Rootkits Fit In

- **Attacker uses a rootkit to gain a stronger foothold on the system**
- **Rootkits aim to prevent or delay discovery by hiding an attacker's resources on a compromised system**
- **Rootkit can also re-enforce an attacker's system access by providing a stealth backdoor**

Attack Scenario - Haxdoor

- **Employee visits a malicious website that exploits an IE 0day**
- **Site installs malware that includes a rootkit**
- **While on the system, the malware steals usernames and passwords, periodically emailing them to an attacker**
- **Malware also installs a backdoor, which the attacker uses to steal confidential documents**
- **Malware goes undiscovered for a long period of time, allowing the attacker to steal large amounts of information**

Attack Scenario - Insider

- **IT worker discovers that he will be fired**
- **He installs a kernel-level rootkit on the web server**
- **After he is fired, the system is audited for backdoors or security holes, but none are found (hidden by rootkit)**
- **Attacker uses access to the web server to steal information, take down site, etc., causing financial loss for his former employer**

How Rootkits Work

How They Work

- **To access files, registry, etc. on system...**
 - **User interacts with GUI or CLI**
 - **Application developer interacts with Win32 API**
 - **Most rootkits are implanted at a much lower level, deep within the operating system**

User-mode vs. Kernel-mode

- Applications run with *user-mode* privileges
 - Cannot access operating system's memory
 - Limited access to other process's memory
 - Limited access to instruction set
- This provides
 - Stability
 - Security

User-mode vs. Kernel-mode

- **Most operating system code and drivers run with *kernel-mode* privileges**
 - Access to all memory
 - Access to all instructions
 - Can directly access system's resources
- **User-mode code usually accesses resources with the Win32 API**
- **Win32 API uses the Native API, which uses kernel-mode system services (system calls) to access resources**

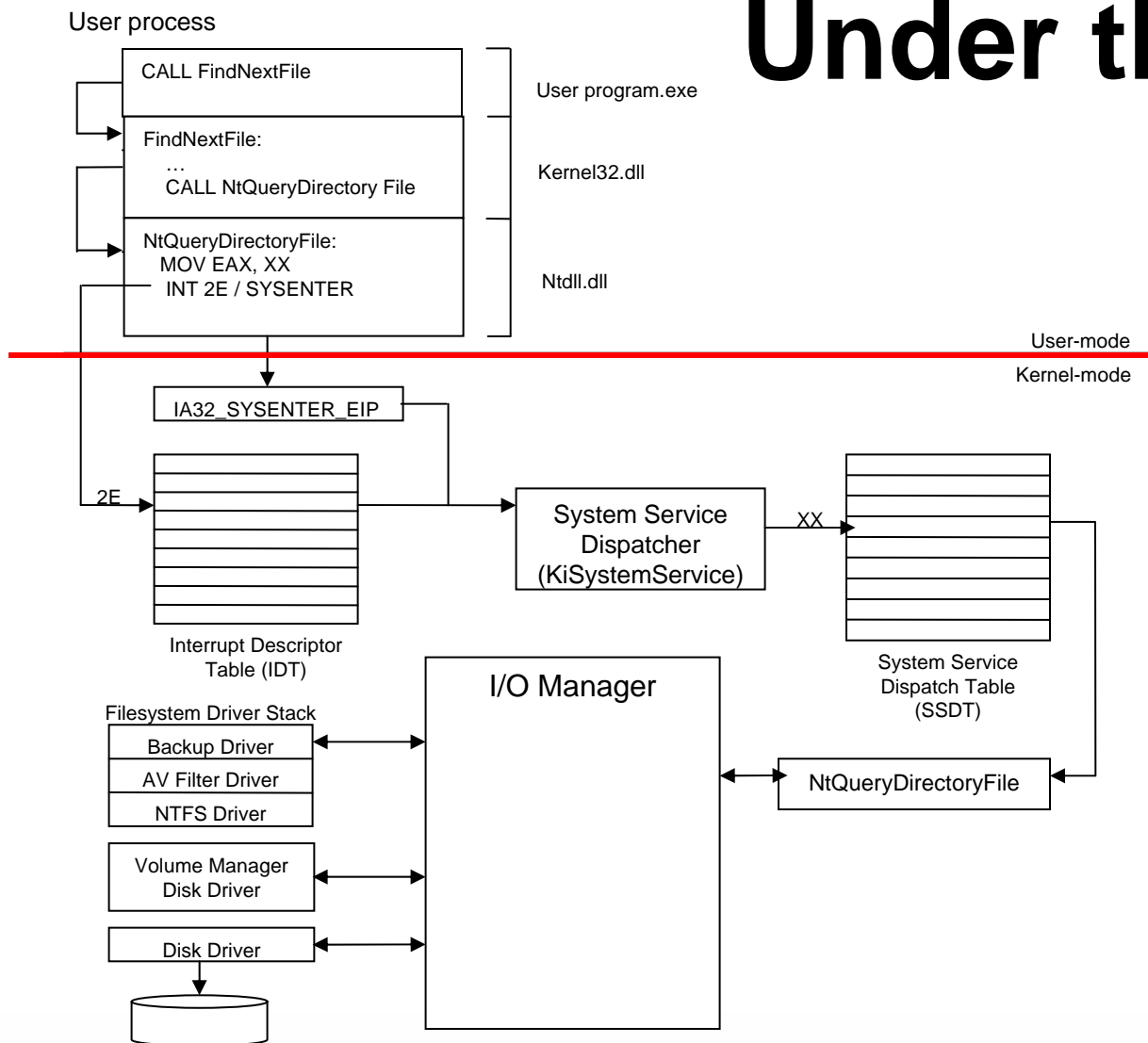
User-mode vs. Kernel-mode

User-mode	Kernel-mode
Limited memory access	Unlimited memory access
Limited instruction set	Full instruction set
Needs to access system resources through kernel	Can directly access system resources

How Rootkits Work

- **Example: Listing files in a directory**
 - User 'dir', Explorer, etc.
 - Win32 Programmer: FindFirstFile() and FindNextFile()
 - Under the hood...

Under the Hood



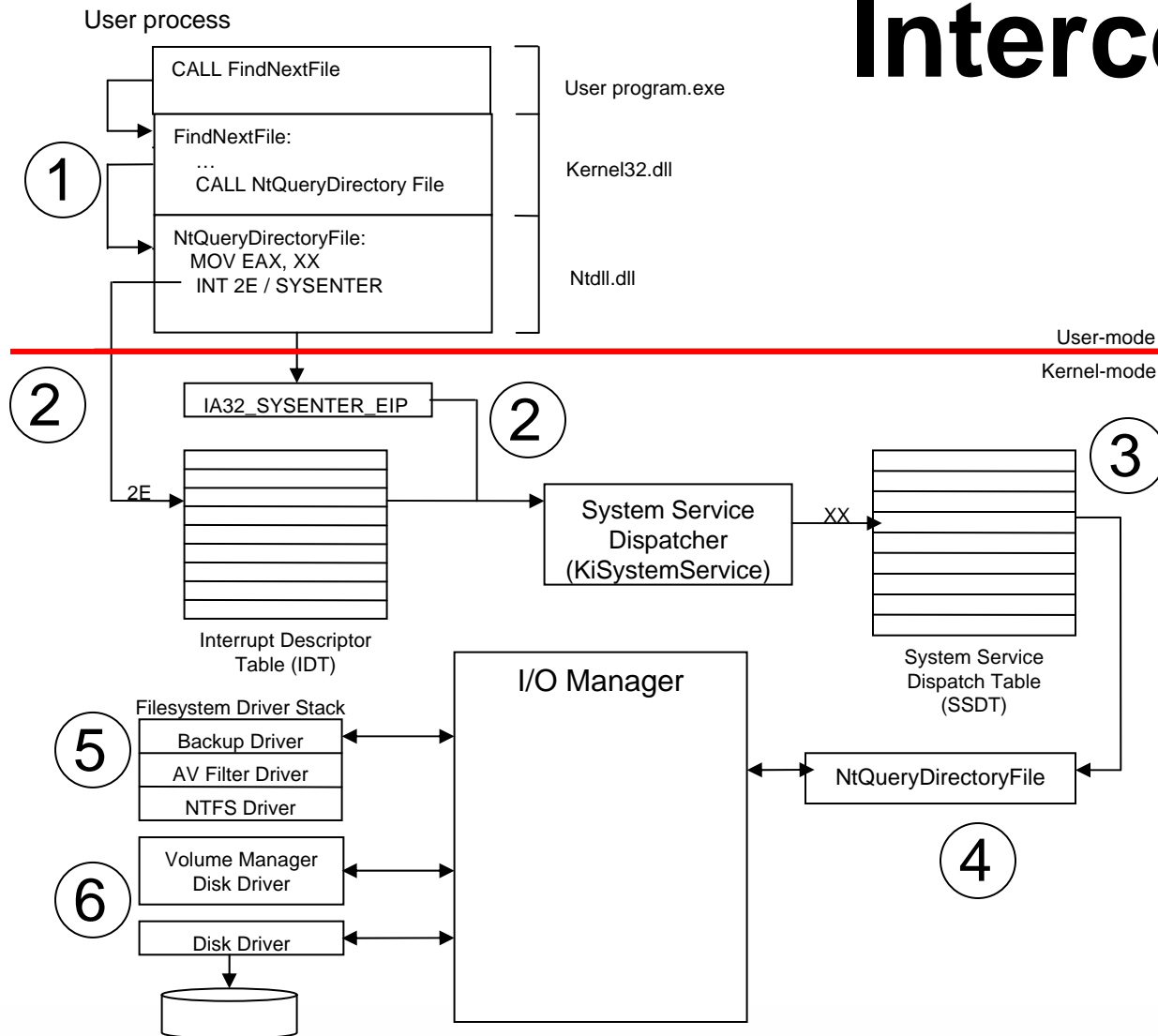
Why So Complex?

- Convenience
- Flexibility
- Portability

Interception

- **Rootkits can intercept requests to:**
 - **Block request**
 - **Alter request**
 - **Fabricate results**
 - **Alter results**
- **Interception is also useful for stealing information**

Interception



1. User-mode hooks
2. IDT / SYSENTER hooks
3. SSDT hooks
4. Kernel code patching
5. Layered driver
6. Driver hooks

User-Mode Interception

- **Pro: Easier to develop code**
- **Con: Easier to detect**

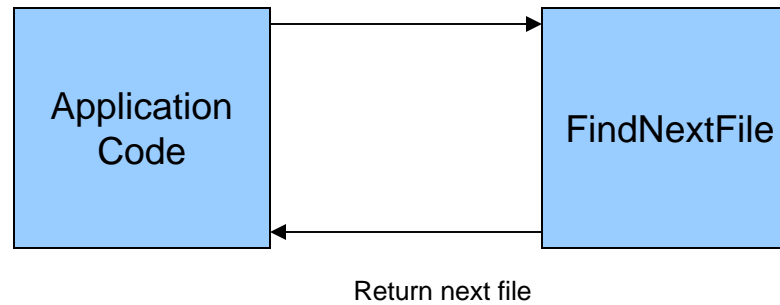
- **Methods:**
 - Import Address Table (IAT) Hooks
 - Export Address Table (EAT) Hooks
 - Inline Hooks
- **Examples: Vanquish, Haxdoor, Hacker Defender (some are hybrids)**

Inline hooking

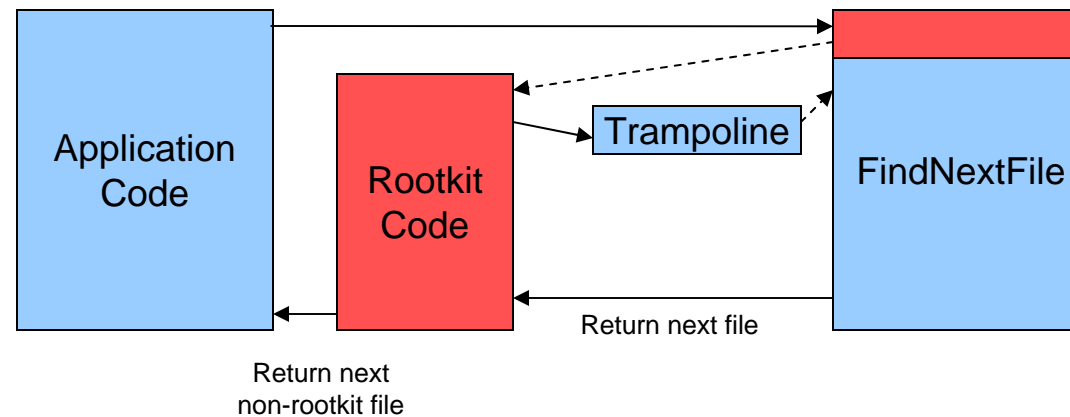
- **Overwrite first few bytes of target function with a jump to rootkit code**
- **Create “trampoline” function that first executes overwritten bytes from original function, then jumps back to original function**
- **When function is called, rootkit code executes**
- **Rootkit code calls trampoline, which executes original function**

Inline hooking

Before:



After:



Installation – User-mode

- In order to hook functions in a given process, rootkit can inject code into process
- Win32 API provides functions for this
 - WriteProcessMemory()
 - CreateRemoteThread() or SetThreadContext()
- Injected code can insert jumps and create trampoline functions

Kernel-mode Interception

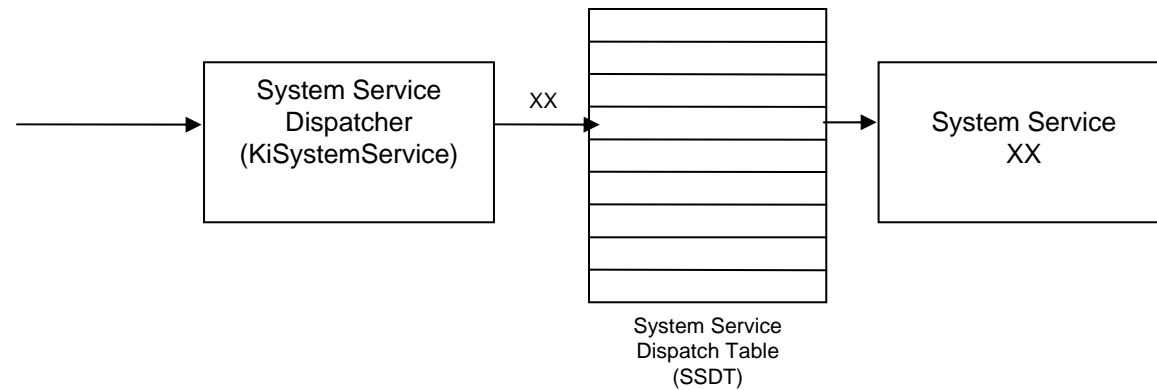
- **Pros:** Can be difficult to detect, many places to intercept
- **Cons:** Complex to implement, can make system unstable
- **Methods:**
 - IDT, SYSENTER, SSDT, driver hooks
 - Layered drivers
 - Code patching

SSDT Hooking

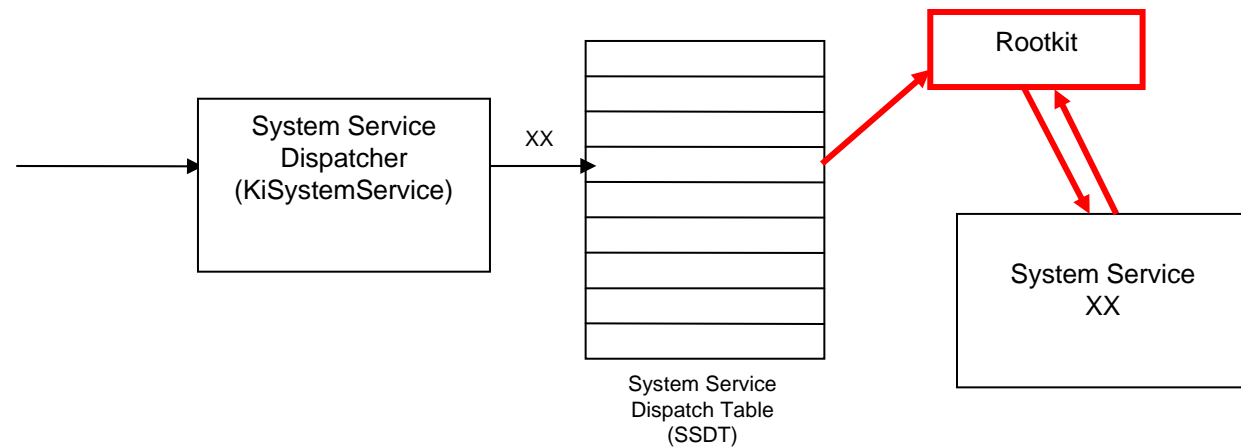
- **System services (system calls) used to access/manipulate:**
 - Filesystem
 - Registry
 - Processes and Threads
 - Memory
- **System Service Dispatch Table (SSDT) has an entry for each system service that contains the service's address**

Example: SSDT Hooking

Before:



After:

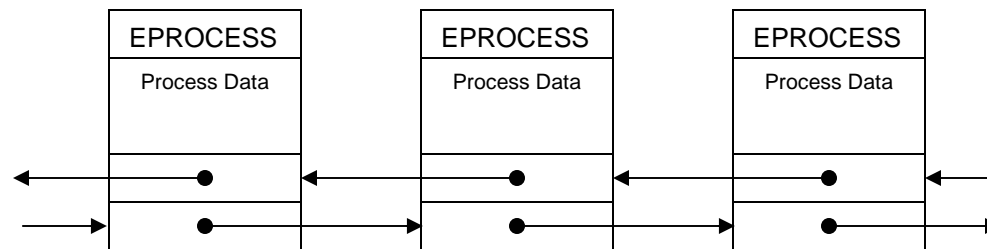


Direct Kernel Object Manipulation

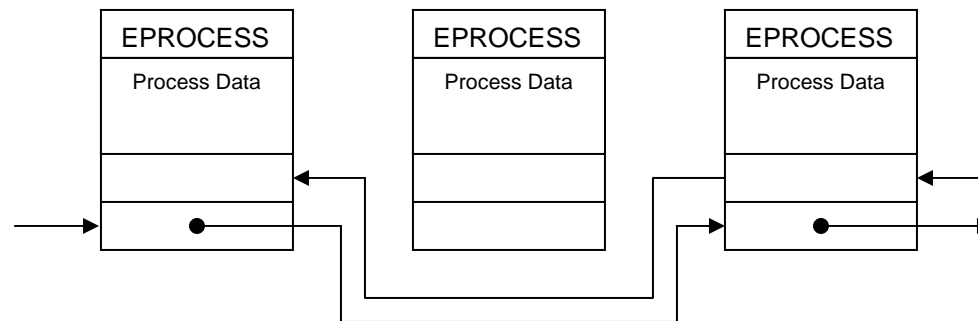
- **Kernel uses data objects to keep track of almost everything**
 - Processes, loaded drivers, etc.
- **Instead of using code to hide resources, manipulate objects**
- **Take advantage of redundancy**
- **Examples: FU, FUTO**

DKOM with Processes

Before:



After:



Installation – Kernel-Mode

- **How an Attacker can inject code into the kernel from an Admin account**
 - Load a driver
 - Manipulate `\Device\PhysicalMemory`
 - Exploit kernel vulnerability

Stealth Backdoors

- **Not usually focused on by rootkit authors**
- **Possibilities**
 - **Steganography – Hide data in TCP packet Fields**
 - **Hide in “normal” traffic**
 - **HTTP, DNS**
 - **Covert**
 - **Can bypass network filtering**



Defending Against Rootkits

Defensive Measures

- **Reactive**
 - Detect Rootkit **AFTER** it has been installed
- **Proactive**
 - Prevent rootkit from being installed
 - Prevent compromise in the first place

Detection

- **Difficult, because...**
 - **Rootkit's goal is to hide**
 - **Usually cannot trust operating system**

Integrity-based Detection

- **Use checksums to monitor system files for changes**
- **Ex. Tripwire**
- **Successful against early rootkits that modified system utilities**
- **Most modern rootkits target memory, so not as successful today**

Signature-based Detection

- **Develop “signatures” for known rootkits**
 - **Sequence of bytes**
- **Scan files / memory for signatures**
- **Cannot detect unknown rootkits**

Hook Detection

- **Most hooks can be detected using heuristics**
 - Jumps at the start of a function
 - Table entries in memory vs. in binary file do not match
- **Examples: VICE, SDTRestore**

Hook Detection

- **False positives**
 - **Some functions appear to be hooked**
 - **Some legitimate software uses hooks**
 - **Personal Firewalls**
 - **Host Intrusion Prevention Systems**

Example: VICE

Hacker
Defender

False
Positives

Hacker
Defender

VICE Console

User Mode Rootkits:

Infected Process	DLL Name	Function	Hook Address	Hooker
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	NtRenameKey	0x7f93fe9	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	NtSetContextThread	0x7f93dc1	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	NtWriteFile	0x7f93d52	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	RtlGetNtVersionNumbers	0x7f93b5e	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	ZwCreateFile	0x7f9488d	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	ZwDeviceIoControlFile	0x7f945f7	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	ZwEnumerateKey	0x7f93e1c	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	ZwEnumerateValueKey	0x7f93f11	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	ZwOpenProcess	0x7f94828	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	ZwQueryDirectoryFile	0x7f93cf0	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	ZwQuerySystemInformation	0x7f93b5e	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	ZwQueryVolumeInformationFile	0x7f94527	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	ZwReadVirtualMemory	0x7f93fe9	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	ZwResumeThread	0x7f93dc1	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ntdll.dll	ZwVdmControl	0x7f93d52	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\kernel32.dll	ReadFile	0x7f93a74	
C:\WINDOWS\Explorer.EXE:312	KERNEL32.dll	GetLastError	0x7c910331	C:\WINDOWS\system32\ntdll.dll
C:\WINDOWS\Explorer.EXE:312	KERNEL32.dll	RtlUnwind	0x7c937a40	C:\WINDOWS\system32\ntdll.dll
C:\WINDOWS\Explorer.EXE:312	KERNEL32.dll	HeapReAlloc	0x7c9179fd	C:\WINDOWS\system32\ntdll.dll
C:\WINDOWS\Explorer.EXE:312	KERNEL32.dll	HeapFree	0x7c91043d	C:\WINDOWS\system32\ntdll.dll
C:\WINDOWS\Explorer.EXE:312	KERNEL32.dll	HeapAlloc	0x7c9105d4	C:\WINDOWS\system32\ntdll.dll
C:\WINDOWS\Explorer.EXE:312	KERNEL32.dll	HeapSize	0x7c9109ed	C:\WINDOWS\system32\ntdll.dll
C:\WINDOWS\Explorer.EXE:312	KERNEL32.dll	DeleteCriticalSection	0x7c91188a	C:\WINDOWS\system32\ntdll.dll
C:\WINDOWS\Explorer.EXE:312	KERNEL32.dll	EnterCriticalSection	0x7c901005	C:\WINDOWS\system32\ntdll.dll
C:\WINDOWS\Explorer.EXE:312	KERNEL32.dll	LeaveCriticalSection	0x7c9010ed	C:\WINDOWS\system32\ntdll.dll
C:\WINDOWS\Explorer.EXE:312	KERNEL32.dll	SetLastError	0x7c910340	C:\WINDOWS\system32\ntdll.dll
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ADVAPI32.dll	EnumServicesStatusA	0x7f9435e	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ADVAPI32.dll	EnumServicesStatusExA	0x7f943c1	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ADVAPI32.dll	EnumServicesStatusExW	0x7f94487	
C:\WINDOWS\Explorer.EXE:312	C:\WINDOWS\system32\ADVAPI32.dll	EnumServicesStatusW	0x7f94421	

Kernel Mode Rootkits:

Infected Object	Function	Hook Address	Rootkit Path
NTOSKRNL.EXE	NtConnectPort	0xe23f2828	

Scan Now Done About...

Example: False Positives (IceSword)

3A7D6

File Dump Plugin View Help

Log

Functions

System Service Descriptor Table

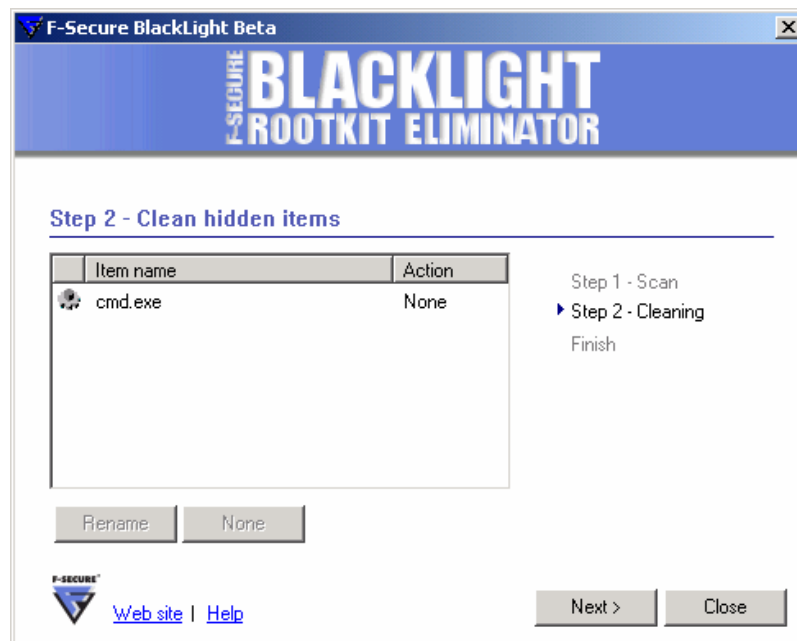
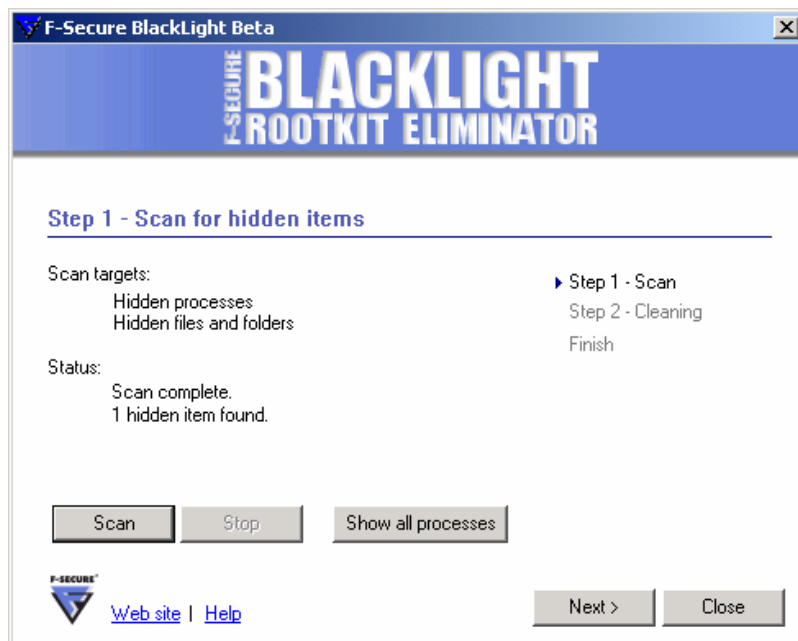
Index	Current Addr	KModule	Original Addr	Name
0xD5	0x80493A5C	{WINNT\System32\ntoskrnl.exe	0x80493A5C	NtSetTimerResolution
0xD6	0xBFE93BA0	csacentr.sys	0x804951C0	NtSetUuidSeed
0xD7	0xF04328CA	csareg.sys	0x80513DE6	NtSetValueKey
0xD8	0xBFE93C30	csacentr.sys	0x804AA372	NtSetVolumeInformationFile
0xD9	0xBFE93CD0	csacentr.sys	0x80490348	NtShutdownSystem
0xDA	0xBFE93D60	csacentr.sys	0x8044F6E4	NtSignalAndWaitForSingleObject
0xDB	0xBFE93E00	csacentr.sys	0x804963C2	NtStartProfile
0xDC	0xBFE93E90	csacentr.sys	0x80496670	NtStopProfile
0xDD	0x804E522E	{WINNT\System32\ntoskrnl.exe	0x804E522E	NtSuspendThread
0xDE	0xBFE93F20	csacentr.sys	0x80525F6A	NtSystemDebugControl
0xDF	0xBFE93FB8	csacentr.sys	0x804E74C2	NtTerminateJobObject
0xE0	0x804E3126	{WINNT\System32\ntoskrnl.exe	0x804E3126	NtTerminateProcess
0xE1	0x804E334A	{WINNT\System32\ntoskrnl.exe	0x804E334A	NtTerminateThread
0xE2	0x804E5550	{WINNT\System32\ntoskrnl.exe	0x804E5550	NtTestAlert
0xE3	0xBFE94050	csacentr.sys	0x805262AA	NtUnloadDriver
0xE4	0xBFE940E0	csacentr.sys	0x805140B4	NtUnloadKey
0xE5	0x804A7624	{WINNT\System32\ntoskrnl.exe	0x804A7624	NtUnlockFile
0xE6	0xBFE94170	csacentr.sys	0x804D2546	NtUnlockVirtualMemory
0xE7	0x804D0B64	{WINNT\System32\ntoskrnl.exe	0x804D0B64	NtUnmapViewOfSection
0xE8	0xBFE94208	csacentr.sys	0x8050296E	NtVdmControl
0xE9	0x8044F8C8	{WINNT\System32\ntoskrnl.exe	0x8044F8C8	NtWaitForMultipleObjects
0xEA	0x804D73DE	{WINNT\System32\ntoskrnl.exe	0x804D73DE	NtWaitForSingleObject
0xEB	0xBFE94298	csacentr.sys	0x80497228	NtWaitHighEventPair
0xEC	0xBFE94330	csacentr.sys	0x804971CA	NtWaitLowEventPair
0xED	0x804AB2BC	{WINNT\System32\ntoskrnl.exe	0x804AB2BC	NtWriteFile
0xEE	0xBFE943C8	csacentr.sys	0x804AB962	NtWriteFileGather
0xEF	0x804C7570	{WINNT\System32\ntoskrnl.exe	0x804C7570	NtWriteRequestData
0xF0	0xBFE61550	csacentr.sys	0x804D4486	NtWriteVirtualMemory
0xF1	0xBFE94458	csacentr.sys	0x804C4980	NtCreateChannel
0xF2	0xBFE944E8	csacentr.sys	0x804C4980	NtListenChannel
0xF3	0xBFE94578	csacentr.sys	0x804C4980	NtOpenChannel
0xF4	0xBFE94608	csacentr.sys	0x80432F0C	NtReplyWaitSendChannel
0xF5	0xBFE946A0	csacentr.sys	0x80432F14	NtSendWaitReplyChannel
0xF6	0xBFE94738	csacentr.sys	0x804C4988	NtSetContextChannel
0xF7	0x80432F1C	{WINNT\System32\ntoskrnl.exe	0x80432F1C	NtYieldExecution

Author:pjf(ustc)

Cross-view Detection

- **Try to look at same data using multiple methods, look for discrepancies in different views**
- **Two Approaches**
 - **Look at multiple places when data is redundantly stored**
 - **Look at same place from high level and low level**
- **Examples: Rootkit Revealer, F-Secure BlackLight, MS Strider GhostBuster**

Example: BlackLight



CMD.exe process hidden by FU rootkit

Combination Tools

- **Each method has strengths and weaknesses, so combine methods**
- **Examples:**
 - **IceSword**
 - **RAIDE**
 - **KProcCheck**

Example: IceSword

**CMD.exe
process
hidden by
FU rootkit**

2DE9A736

File Dump Plugin View Help

Process: 36

ImageName	PID	PathName	Base Priority	EPROCESS	State	Mem Usage	Peak Mem ...
System	8	NT OS Kernel	8	0x814A1020	Ready	140k	848k
SMSS.EXE	152	C:\WINNT\system32\SMSS.EXE	11	0x811A6D60	Idle	164k	2004k
CSRSS.EXE	176	C:\WINNT\system32\CSRSS.EXE	13	0x81116D20	Ready	1236k	1780k
WINLOGON.EXE	196	C:\WINNT\system32\WINLOGON.EXE	13	0x810D5D60	Ready	728k	8820k
SERVICES.EXE	228	C:\WINNT\system32\SERVICES.EXE	9	0x811F4C60	Ready	2964k	38692k
LSASS.EXE	248	C:\WINNT\system32\LSASS.EXE	9	0x811C9C20	Ready	1076k	5028k
svchost.exe	436	C:\WINNT\system32\svchost.exe	8	0x81121D60	Ready	1512k	3876k
jusched.exe	452	C:\Program Files\Java\j2re1.4.2_07\bin\jusched.exe	8	0x86BF8520	Idle	1856k	1916k
spoolsv.exe	468	C:\WINNT\system32\spoolsv.exe	8	0x8122AD60	Idle	1512k	4568k
csacontrol.exe	528	C:\Program Files\Cisco Systems\CSAgent\bin\csacontrol.exe	8	0x80092D60	Idle	828k	5484k
svchost.exe	564	C:\WINNT\system32\svchost.exe	8	0x86CC64A0	Idle	1228k	6992k
leventmgr.exe	572	C:\Program Files\Cisco Systems\CSAgent\bin\leventmgr.exe	8	0x86CC6820	Ready	5508k	8480k
Framework...	596	C:\Program Files\Network Associates\Common Framework\...	8	0x8776B4E0	Ready	7672k	10888k
Mcshield.exe	788	C:\Program Files\Network Associates\VirusScan\Mcshield.exe	13	0x84AF5020	Ready	4624k	23728k
VsTskMgr.exe	804	C:\Program Files\Network Associates\VirusScan\Vstskmgr.exe	8	0x84BC0D60	Ready	2084k	3520k
regsvcs.exe	892	C:\WINNT\system32\regsvcs.exe	8	0x82AB4660	Idle	260k	1000k
mstask.exe	896	C:\WINNT\system32\mstask.exe	8	0x85468B60	Idle	1272k	3132k
naPrdMgr.exe	928	C:\PROGRA~1\NETWOR~1\COMMON~1\naPrdMgr.exe	8	0x84D42D60	Idle	2352k	10660k
WinMgmt.exe	944	C:\WINNT\system32\wbem\WinMgmt.exe	8	0x85B1F7E0	Idle	628k	5424k
WinRAR.exe	956	C:\Program Files\WinRAR\WinRAR.exe	8	0x879AC020	Ready	1088k	5900k
svchost.exe	996	C:\WINNT\system32\svchost.exe	8	0x84BD4A40	Idle	1204k	4468k
inetinfo.exe	1060	C:\WINNT\system32\inetrv\inetinfo.exe	8	0x863509A0	Ready	1764k	23888k
shstat.exe	1244	C:\Program Files\Network Associates\VirusScan\shstat.exe	8	0x863532C0	Ready	484k	3680k
UpdaterUI.exe	1252	C:\Program Files\Network Associates\Common Framework\...	8	0x870AB8C0	Ready	1696k	3932k
MDM.EXE	1384	C:\WINNT\system32\MDM.EXE	8	0x8748FD60	Ready	1272k	2988k
CMD.EXE	1420	C:\WINNT\system32\CMD.EXE	8	0x833439C0	Idle	1288k	1288k
explorer.exe	1496	C:\WINNT\explorer.exe	8	0x822F8AE0	Ready	4160k	23228k
IceSword.exe	1540	C:\Documents and Settings\administrator\Desktop\Ice...	8	0x833133E0	Ready	6060k	6088k
svchost.exe	1564	C:\WINNT\system32\svchost.exe	8	0x808A4020	Idle	852k	5324k
TASKMGR.EXE	1624	C:\WINNT\system32\TASKMGR.EXE	13	0x80399D60	Ready	1068k	2580k
mconsol.exe	1640	C:\Program Files\Network Associates\VirusScan\mconsol.exe	8	0x82956020	Ready	416k	4028k
jucheck.exe	1656	C:\Program Files\Java\j2re1.4.2_07\bin\jucheck.exe	8	0x843CF8E0	Idle	2120k	3312k
okclient.exe	1728	C:\Program Files\Cisco Systems\CSAgent\bin\okclient.exe	13	0x878DEAE0	Idle	2292k	6476k
MSPAINT.EXE	1740	C:\WINNT\system32\MSPAINT.EXE	8	0x855B76E0	Idle	1208k	22380k
CMD.EXE	1772	C:\WINNT\system32\CMD.EXE	8	0x8489E700	Idle	764k	1388k

Author:pj(ustc)

Example: KProcCheck

```

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\cries\Desktop\rootkits\KProcCheck>kproccheck.exe
KProcCheck Version 0.2-beta2 Proof-of-Concept by SIG^2 (www.security.org.sg)

Usage: kproccheck.exe -p      show kernel active process list.
       kproccheck.exe -s      show scheduler thread list.
       kproccheck.exe -o      show processes by traversing Handle Table list.
       kproccheck.exe -d      show kernel module list.
       kproccheck.exe -t      show hooked SDT entries.
       kproccheck.exe -g      show hooked GDI SDT entries.
       kproccheck.exe -u      unload support driver.

C:\Documents and Settings\cries\Desktop\rootkits\KProcCheck>
  
```

```

C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Remote_3\Desktop\KProcCheck>KProcCheck.exe -o
KProcCheck Version 0.2-beta2 Proof-of-Concept by SIG^2 (www.security.org.sg)

Process list by traversal of HandleTableList

4      -      System
192    -      alg.exe
252    -      cmd.exe      --[Hidden]--
468    -      ati2evxx.exe
480    -      ccProxy.exe
504    -      DefWatch.exe
632    -      smss.exe
732    -      csrss.exe
760    -      winlogon.exe
804    -      services.exe
816    -      lsass.exe
936    -      KProcCheck.exe
972    -      svchost.exe
1036   -      svchost.exe
1132   -      svchost.exe
1204   -      SndSrv.exe
1228   -      svchost.exe
1300   -      wdfmgr.exe
1372   -      svchost.exe
1400   -      RtvsScan.exe
1408   -      svchost.exe
1540   -      SymSPort.exe
1576   -      ccSetMgr.exe
1604   -      ccEvtMgr.exe
1732   -      spoolsv.exe
1836   -      vtserver.exe      --[Hidden]--
2652   -      hxdef100.exe
2864   -      explorer.exe
3024   -      Hcontrol.exe
3040   -      atiptaxx.exe
3048   -      SynTPLPr.exe
3056   -      SynTPEnh.exe
3064   -      AGRSMMSG.exe
3072   -      PwdProt.exe
3080   -      FDBkgr.exe
3192   -      PPControl.exe
3200   -      PPMemCheck.exe
3208   -      CookiePatrol.exe
3216   -      ccApp.exe
3224   -      UPTray.exe
3232   -      msmsgs.exe
3268   -      OLSNT40.EXE
3608   -      notepad.exe
3684   -      ATKOSD.exe
3860   -      ViceConsole.exe

Total number of processes = 45
C:\Documents and Settings\Remote_3\Desktop\KProcCheck>
  
```

CMD.exe and
hxdef100.exe
processes hidden by
Hacker Defender
rootkit

Removal

- **Best solution is to rebuild system**
- **Clean the infection**
 - **Remember, OS cannot be trusted!**
 - **Either disable rootkit or boot with clean CD, and remove rootkit's resources**
 - **Need backups or baselines to verify integrity of system files and data**

Anti-detection

- **Rootkit authors respond to detection tools:**
 - **Exploit weaknesses in tool to evade detection – FUTO**
 - **Use signatures to detect detectors – Hacker Defender Gold commercial rootkit**
 - **When detector is detected, either disable rootkit or patch detector**

DEMO (After talk)

- **FUTO vs. BlackLight**
- **FUTO vs. KProcCheck**

Prevention

- **Prevent compromise in the first place!**
- **Good security practices**
 - **System hardening**
 - **Patch Management**
 - **Up to date Anti-virus**
 - **Least Privileges**
 - **Periodic auditing of critical systems**

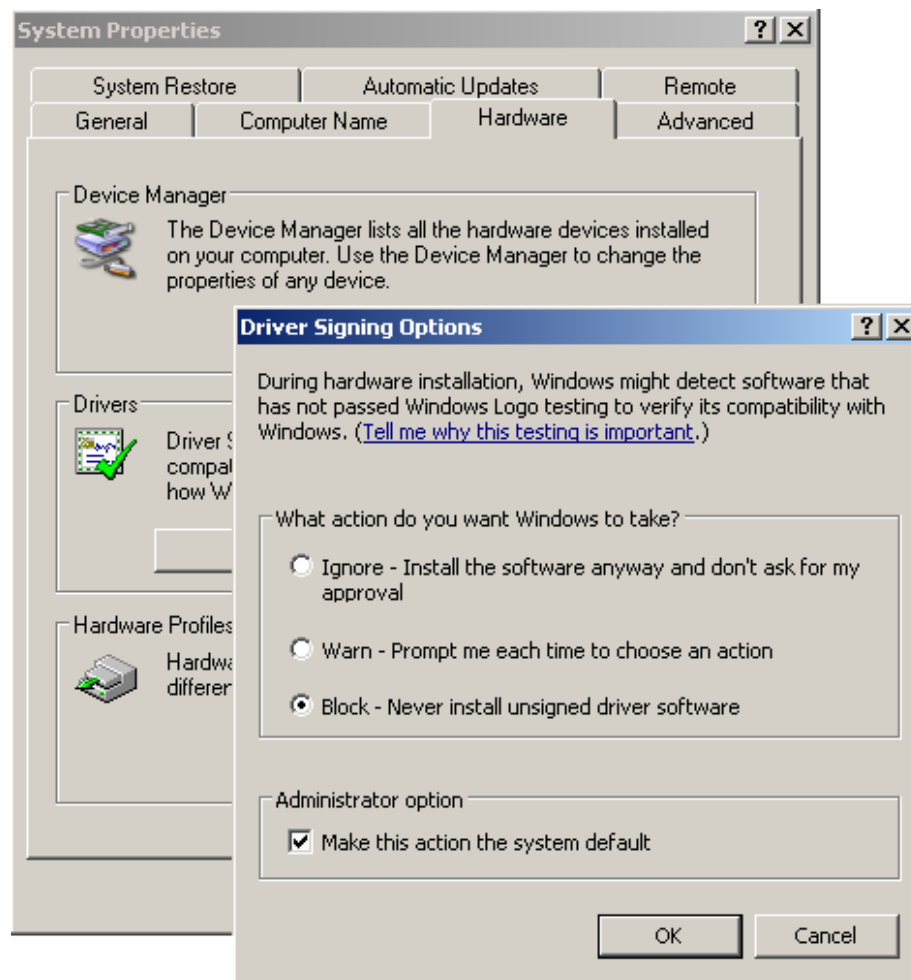
Stages of An Attack

- 1. Vulnerability in a system is discovered <- IDS**
- 2. Vulnerability is exploited to gain access to the system <- Patch Management**
- 3. Attacker gains a foothold on the system by escalating privileges, installing backdoor, etc. <- Host-based security software, least privilege**
- 4. Attacker utilizes system access to steal information, launch other attacks, etc. <- filtering, segmentation**
- 5. Compromise is discovered, and incident response is executed <- Periodic auditing**

Prevention

- **Rootkit-specific**
 - **Disable user-mode access to physical memory**
 - **Do not allow new drivers to be loaded, or only allow digitally signed drivers**
 - **Monitor interactions between processes**

Digitally Signed Drivers

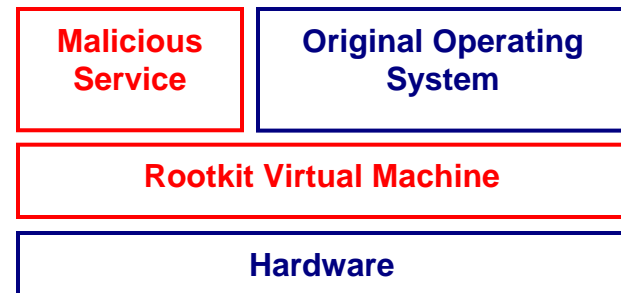


Host Intrusion Prevention Systems

- **Prevent new drivers from being loaded**
- **Protect Physical Memory**
- **Detect and prevent buffer overflow exploits**
- **Restrict interactions between processes such as process injection**

Future of Rootkits

- **Subvert OS at boot**
 - Like boot sector virus
 - eEye BootRoot
 - SubVirt
- **Hardware / Firmware**
 - NGS ACPI BIOS Rootkit
 - NIC Card firmware
- **Hardware-based Detection**
 - Copilot



SubVirt



Thank You!

Questions? Comments?

Contact Info:

412-661-5700 x258

Chris.Ries@vigilantminds.com