Rootkit Technologies

Robert Hensing
FIRST TC 2004

- If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle.
  - Sun Tzu

Agenda

- Rootkit Introduction
- Rootkit methodologies
- Most popular Rootkits
- Rootkit detection
- Finding more information

Rootkit Introduction

- What is a Rootkit?
- Attack scenario
- Method of infection
- User Mode Rootkits
- Kernel Mode Rootkits

What is a Rootkit?

- Tool to hide your presence, maintain system/root/admin privileges and perform activities without detection.
- What does that mean????
- Well….

A Rootkit can:

- Hide processes
- Hide files
- Hide drivers
- Hide ports and network connections
- Install a backdoor listener for future access to the system
- Add Privileges to Tokens
- Add Groups to Tokens
- Manipulate the Event Viewer
- Basically, do anything it is programmed to do

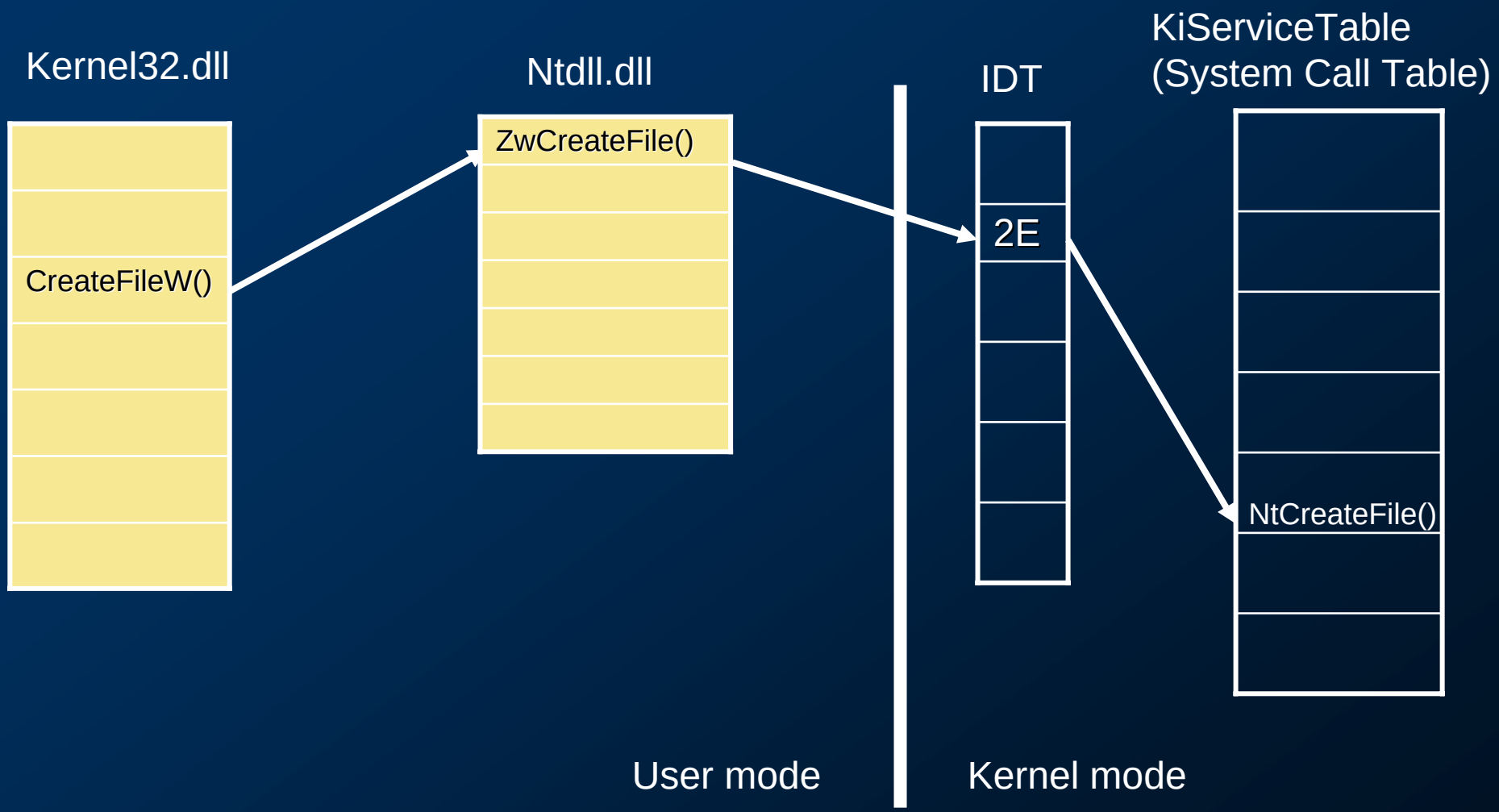
Attack Scenario

- Attacker gains elevated access to computer system
- Attacker installs a Rootkit
- Rootkit hides itself, everything else the hacker wants and provides covert channel for control/management
- Attacker is able to use the system for whatever they want with little risk of detection

State of the Rootkit

- Originally, rootkits were little more than Trojan programs (e.g. ps, ls, netstat)
- Sophisticated rootkits filter data going in and/or out
  - Hook system functions in the kernel
  - Modify key data structures in memory
  - Hook user mode functions in kernel32.dll & ntdll.dll

Oh Where Oh Where Are My Rootkit Hooks



Rootkits can hook any function and/or manipulate any objects in memory.

How does this thing get in???

- A rootkit is nothing more than software. Applications and/or drivers. I just happens to be software that is unwanted and unauthorized.
- There are many methods of installation (loaders).
- To be installed, the install code needs either admin/system level access, or user access with path to elevate (like insecured reg keys, etc), depending on sophistication level of the installer/loader.

Initial Vector

- Weak passwords
- Arbitrary code execution due to:
  - Buffer overflows
  - Incorrectly secured registry keys for privileged apps
  - Web/email exe's, insecure zones, etc.
  - Internet Explorer 0-days
- Social engineering (hack the human)
- Physical access
- Island hopping from other compromised systems

User Mode Rootkits

- Modify programs and libraries
- Hook (redirect/MITM) usermode API libraries:
  - Lot’s of useful APIs in kernel32.dll & ntdll.dll to play with
  - http://rootkit.host.sk/knowhow/hookingen.txt
- User Mode rootkit drawback: Hookers easily detectable from kernel mode

Kernel Mode Rootkits

- Hook/redirect kernel functions- SDT/SST/KiServiceTable/W32pServiceTable
- Hook interrupt – Int 2E – get/filter every function
- Direct Kernel Object Manipulation (DKOM) –
  - EPROCESS FLINK/BLINK – OS schedules threads, not processes
  - Tokens – add groups, change SIDs, rights, etc.
  - Rewrite scheduler list-
    - KiWaitInListHead
    - KiWaitOutListhead
    - KiDispatcherReadyListHead
- Kernel Mode drawbacks: None other than complexity – can be virtually undetectable – esp. in cases of DKOM

Future? Combo Malware

- Worm +
- Rootkit +
- Polymorphic +
- Alters BIOs - reinstallation

Some Popular Rootkits

- Hacker Defender 1.0.0
- Aphex 2003 Rootkit
- FU
- HE4Hook
- Vaniquish

Hacker Defender 1.0.0

- User mode/kernel mode
- Very popular (and lots of features)
- Hides:
  - Files
  - Dirs
  - Processes
  - Registry keys
  - Services
  - Drivers
- Starts programs when a user logs on
- Provides a backdoor listener (on ALL ports)
- Provides redirector functionality

Hacker Defender 1.0.0

- Hooks:
  - Kernel32.ReadFile
  - Ntdll.NtQuerySystemInformation (class 5 a 16)
  - Ntdll.NtQueryDirectoryFile
  - Ntdll.NtVdmControl
  - Ntdll.NtResumeThread
  - Ntdll.NtEnumerateKey
  - Ntdll.NtEnumerateValueKey
  - Ntdll.NtReadVirtualMemory
  - Ntdll.NtQueryVolumeInformationFile
  - Ntdll.NtDeviceIoControlFile
  - Ntdll.NtLdrLoadDll
  - Ntdll.NtOpenProcess
  - Ntdll.NtCreateFile
  - Ntdll.NtLdrInitializeThunk
  - WS2_32.recv
  - WS2_32.WSARecv
  - Advapi32.EnumServiceGroupW
  - Advapi32.EnumServicesStatusExW
  - Advapi32.EnumServicesStatusExA
  - Advapi32.EnumServicesStatusA

Hacker Defender 1.0.0

- Detection (Public): VICE, Patchfinder2
- Detection (Microsoft): NFF, TKTracker

AFX (2003)

- Aka Aphex
- Uses DLL injection
- Hides:
  - Processes
  - Files
  - Folders
  - Registry keys
  - Ports
- Detection (Public): VICE, Patchfinder2
- Detection (Microsoft): NFF, TKTracker

FU

- DKOM
- Very popular
- Hides:
  - Processes (unlinks from PsActiveProcessList)
  - Drivers
- Changes tokens, process privs, AUTH_ID

- Detect: klister

HE4Hook

- Modifies kernel SDT
- Detection (Public): Patchfinder, VICE
- Detection (Microsoft): NFF, TKTracker, RKDS

Vanquish

- DLL-Injection based rootkit that hides files, folders, registry entries and logs passwords
- Hooks:
  - CreateProcess(AsUser)A/W
  - FreeLibrary
  - FindFirstFileExW, FindNextFileW
  - RegCloseKey, RegEnumKeyA/W, RegEnumKeyExA/W
  - RegEnumValueA/W, RegQueryMultipleValuesA/W
  - EnumServicesStatusA/W
  - LogonUserA/W, WlxLoggedOutSAS
  - DeleteFileA/W, RemoveDirectoryA/W
  - SetLocalTime, SetTimeZoneInformation
  - SetSystemTimeAdjustment, SetSystemTime
- Detection (Public): VICE, Patchfinder2
- Detection (Private): NFF, TKTracker

Rootkit Detection

- Current Public Tools:
  - VICE
  - Patchfinder2
  - Klister
- Current Microsoft Tools:
  - NFF
  - RKDS
  - TKTracker

VICE

- Detects:
  - User mode API hooks
  - Kernel mode hooks
- Will usually find anyone hooking in the system

- http://www.rootkit.com/project.php?id=20

Patchfinder2

- Detects many rootkits – including most popular ones
- Very effective if baseline is created when system is known good
- Counts instructions for system functions and compares against baseline
- * Does not detect rootkits using DKOM
- http://www.rootkit.com/project.php?id=15

Klister

- Dumps:
  - IDT
  - SDT
  - Dispatcher structures
- Will detect FU
- http://www.rootkit.com/project.php?id=14

RKDS

- Our first rootkit detection tool
  - Does 3 different checks
    - Usermode vs. Kernelmode process list comparison
    - Checks integrity of SDT
    - Kernel driver verification
- Problems
  - Driver checks require knowledge of specific location in memory
    - It changes from hotfix to hotfix / service pack to service pack
  - Old tool – miscreants have moved on

TKTracker

- Innovative approach to detection
- Loads via appinit_dlls at boot time (requires reboot)
- Can identify hidden processes, drivers, services and optionally block them from loading
- Very verbose logging
- Not OS dependant like RKDS

NFF

- Our newest tool
- Finds file-hiding rootkits only
- Dumps the MFT raw and for each file / folder entry makes 4 different Win32 API calls to see if the API knows about the file / folder.
  - If not – this implies the file / folder is being hidden by a file-hiding rootkit
- Highly effective. ☺