Desktop Forensics: Windows

Part II.A. Techniques and Tools: Computer Forensics

CSF: Forensics Cyber-Security
Fall 2015
Nuno Santos
Summary

- Windows forensics
  - Windows boot sequence
  - Relevant Windows data structures
  - Artifacts of user activities
2. Operating system forensics

1. Storage stack forensics

Software

Operating System

Hardware

CPU
RAM
I/O

Application
Application
Application
Importance of operating system forensics

- Ultimately, in a forensic examination, we’re investigating the actions of a **person**.

- Almost every **event** or **action** on a system is the result of a user either doing something (or not doing something).

- Many of such events introduce changes to the system state that are supervised by the **operating system (OS)**.

- **OS forensics** helps understand how system changes correlate to events resulting from the actions of somebody in the real world.
Operating systems we will be focusing on

- **Desktop platforms**: Windows
- **Server platforms**: Linux
- **Mobile platforms**: Android

Today!

We will learn methods and techniques to help us extract and interpret data of investigative value from computers running Windows operating system.
There are many versions of windows out there

Some of the older versions are outdated and are no longer used:
- Windows 9x, NT, ME, 2000

In home, corporate, and government environments it’s far more likely to find newer versions
- >= Windows XP

We will cover the newer recent versions of Windows, focusing on their common features
Windows boot sequence
Windows startup: Why relevant for forensics?

1. Interrupt the boot process to view and document the CMOS configuration

2. Explain which files were altered in the startup process
   - E.g., if an evidentiary system was accidentally booted, demonstrate that no user-created files were modified

3. Determine which version of the OS was running and when was installed

4. Examine the startup process for signs of tampering
   - E.g., important when investigating malware
A recent examination of a server and analysis of its web logs confirmed that the system had indeed been compromised. The logs provided a suspicious IP address, the operating system used in the compromise (Windows XP), and the suspect’s web browser. The IP address led investigators to an ISP, and eventually to a suspect in the intrusion.

However, the suspect denied all involvement in the compromise and stated that this computer was running Windows 98 (as has always been the case). This was of course discouraging news for investigators, who were sure they had their man.

Investigators began forensics examination of the suspect’s computer. A search of the hard drive revealed a deleted boot.ini file that appeared to have been deleted mere days after the compromise of the web server, clearly showing that Windows XP Professional had been installed on the system, thereby punching a hole in the suspect’s story.
All NTFS computers perform the following steps when the computer is turned on:

1. Power-on self test (POST)
2. Initial startup
3. Boot loader
4. Hardware detection and configuration
5. Kernel loading
6. User logon

Windows-specific code
Phase 1: Power-On Self Test (POST)

1. Power supply performs self-test
2. CPU loads the ROM BIOS code
3. ROM BIOS performs a basic test of central hardware
4. BIOS checks adapters requiring own ROM BIOS routines
5. ROM BIOS checks if this is a cold boot or a warm boot
   - Cold boot (startup from a powered-down state): a full POST
   - Warm boot (restart of a system that is already on): the memory test portion of the POST is switched off
6. POST tests the video card and video memory, and displays configuration information or any errors
7. BIOS reads configuration information stored in CMOS
Phase 2: Initial startup

- The BIOS examines the disk for a master boot record (MBR).

- With a valid MBR loaded into memory, the BIOS transfers control of the boot process to the partition loader code.
Phase 3: Boot loader

- The NTLDR system file controls loading of Windows

1. Initial boot-loader phase:
   - NTLDR switches the CPU to protected mode and turns memory paging on
   - Loads file system drivers to allow file loading from various file systems

2. Operating system selection:
   - If BOOT.INI exists and contains entries multiple OS, NTLDR stops booting, displays a menu of choices, and waits the user to make a selection
   - User can press F8 to display various boot options (e.g., “Safe Mode”)

An example of a boot.ini file:

```
[boot loader]
timeout=40
default=multi(0)disk(0)rdisk(0)partition(1)\WINDOWS

[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINDOWS="Microsoft Windows XP Professional" /fastdetect
```
Phase 3: Boot loader (cont.)

- Boot menu example: User can select from various boot options by pressing F8 during the boot process (same for >=Win XP)
Phase 4: Hardware detection and configuration

1. NTLDR locates and loads the DOS-based NTDETECT.COM program to perform hardware detection.

2. If multiple hardware profile, NTLDR will stop at this point and display the Hardware Profiles/Configuration Recovery menu.

3. After the user selects a hardware configuration, NTLDR begins loading the XP kernel (NTOSKRNL.EXE).
Phase 5: Kernel loading

1. NTOSKRNL goes through two phases in its boot process:
   - Phase 0: The hardware abstraction layer (HAL) is loaded (hal.dll) and called to prepare the interrupt controller
   - Phase 1: All executive subsystems are reinitialized (e.g., cache manager, process manager, I/O manager)
2. I/O Manager starts loading all the system driver files
3. Win32k.sys switches the screen into graphics mode
4. Services subsystem starts services marked as Auto Start
5. Once all devices and services are started, the boot is deemed successful, and this configuration is saved as the last known good configuration
Phase 6: User logon

- The WINLOGON.EXE file starts the logon process
  - Login manager responsible for all login and logout procedures
- The Local Security Authority (LSASS.EXE) process displays the logon dialog box
Contamination concerns with Windows XP

- When you start a Windows XP NTFS workstation, several files are accessed immediately
  - The last access date and time stamp for the files change to the current date and time

- May destroy any potential evidence
  - E.g., that shows when a Windows workstation was last used

- Determining which files are changed upon startup and shutdown can be done using some forensic tools
  - [http://forensicswiki.org/wiki/Files_changed_at_boot:Windows_XP](http://forensicswiki.org/wiki/Files_changed_at_boot:Windows_XP)
Relevant Windows data structures
Relevant Windows data structures

- NTFS (covered in previous classes)
- Windows Registry
- Windows Event Log
The Registry is the heart and soul of Windows OSes and a wealth of information can be recovered:

- System configuration
- Devices on the system
- User names
- Personal settings and browser preferences
- Web browsing activity
- Files opened
- Programs executed
- Passwords
Microsoft defines the Registry thus:

“A central hierarchical database used in Microsoft Windows 9x, Windows CE, Windows NT, and Windows 2000 used to store information necessary to configure the system for one or more users, applications and hardware devices.”

https://support.microsoft.com/en-us/kb/256986
Virtually everything done in Windows refers to or is recorded into the Registry

The RegMon program can be used to display registry activity in real time.

Registry access barely remains idle: the registry is referenced in one way or another with every action taken by the user.
Learning the registry’s history helps understand its structure

The Registry was first introduced with Windows 95

The Registry replaces configuration files used in MSDOS:
- config.sys: to load device drivers and the
- autoexec.bat: to run startup programs and set env variables

It also replaces initialization (.ini) files introduced in Win 3.0
- win.ini and system.ini store user settings and OS parameters
Problems overcome by the Registry

- Proliferation of INI files
- Slow access
- No standards
- Fragmented
The Registry can be seen as a unified file system
A **hive** is a logical group of keys, subkeys, and values in the registry that has a set of supporting files containing backups of its data.

Each time a new user logs on, a new hive is created for that user with a separate file for the user profile:

- User's app settings, desktop, environment, network connections, and printers
- User profile hives are located under the HKEY_USERS key

'**HKEY**' is an abbreviation for Handle to a Key
Root key functions

- **HKEY_LOCAL_MACHINE (HKLM)**
  - Contains system-wide hardware settings and configuration information (e.g., list of drives mounted on the system)

- **HKEY_USERS (HKU)**
  - Contains the root of all user profiles that exist on the system

- **HKEY_CLASSES_ROOT (HKCR)**
  - Ensures the correct program opens when executed in Windows Explorer

- **HKEY_CURRENT_USER (HKCU)**
  - Contains the profile (settings) of the user who is currently logged in

- **HKEY_CURRENT_CONFIG (HCU)**
  - Information about the HW profile used by the computer during start up
Hive’s supporting files

- Hives have sets of supporting files
  - Most of them located in: `%SystemRoot%\System32\Config`
  - These files are updated each time a user logs on

<table>
<thead>
<tr>
<th>Registry hive</th>
<th>Supporting files</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKEY_CURRENT_CONFIG</td>
<td>System, System.alt, System.log, System.sav</td>
</tr>
<tr>
<td>HKEY_CURRENT_USER</td>
<td>Ntuser.dat, Ntuser.dat.log</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\SAM</td>
<td>Sam, Sam.log, Sam.sav</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\Software</td>
<td>Software, Software.log, Software.sav</td>
</tr>
<tr>
<td>HKEY_LOCAL_MACHINE\System</td>
<td>System, System.alt, System.log, System.sav</td>
</tr>
</tbody>
</table>
## Some important hives

<table>
<thead>
<tr>
<th>Filename</th>
<th>Location</th>
<th>Content</th>
</tr>
</thead>
</table>
| ntuser.dat  | \Documents and Settings \user account | Protected storage area for user  
Most Recently Used (MRU) files  
User preference settings |
| Default     | \Windows\system32\config        | System settings                                                          |
| SAM         | \Windows\system32\config        | User account management and security settings                           |
| Security    | \Windows\system32\config        | Security settings                                                        |
| Software    | \Windows\system32\config        | All installed programs and their settings                                |
| System      | \Windows\system32\config        | System settings                                                          |
### Registry data types

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REG_BINARY</td>
<td>Raw binary data</td>
</tr>
<tr>
<td>REG_DWORD</td>
<td>Data represented as a 32-bit (4-byte) integer</td>
</tr>
<tr>
<td>REG_SZ</td>
<td>A fixed-length text string</td>
</tr>
<tr>
<td>REG_EXPAND_SZ</td>
<td>A variable-length data string</td>
</tr>
<tr>
<td>REG_MULTI_SZ</td>
<td>Multiple strings, separated by a space, comma, or other delimiter</td>
</tr>
<tr>
<td>REG_NONE</td>
<td>No data type</td>
</tr>
<tr>
<td>REG_QWORD</td>
<td>Data represented by a 64-bit (8-byte) integer</td>
</tr>
<tr>
<td>REG_LINK</td>
<td>A Unicode string naming a symbolic link</td>
</tr>
<tr>
<td>REG_RESOURCE_LIST</td>
<td>A series of nested arrays designed to store a resource list</td>
</tr>
<tr>
<td>REG_RESOURCE_REQUIREMENTS_LIST</td>
<td>A series of nested arrays designed to store a device driver’s list of possible hardware resources</td>
</tr>
<tr>
<td>REG_FULL_RESOURCE_DESCRIPTOR</td>
<td>A series of nested arrays designed to store a resource list used by a physical hardware device</td>
</tr>
</tbody>
</table>
Relevant Windows data structures

- NTFS (covered in previous classes)
- Windows Registry
- Windows Event Log
Whenever an event, such as a user logging on or off, occurs, the operating system logs the event.

An event can be any occurrence that the OS or a program wants to keep track of or alert the user about.

Windows has a centralized log service to allow apps and OS to report events that have taken place:
- Application (example: Database message)
- System (example: driver failure)
- Security (example: Logon attempt, file access)
Structure of the Event Log

- The Event Log can be seen using a specific system tool.
Events have a specific format and meaning
Example of detailed event tracking

- Detailed Event tracking can include the following events:
  - #528 – Successful Login (The user authenticate to the system)
  - #592 – A new process has been created (application is launched)
  - #560 – Object Open (a file is requested)
  - #567 – Object Access (the file is modified and saved)
  - #564 – Object Deleted
  - #562 – Handle Closed (the file has been closed)
  - #593 – A Process Has Exited (the application was terminated)
Artifacts of user activities
Where to find information about user activities

- **Volatile information**
  - Open network connections
  - Running processes
  - ...

- **Non-volatile information**
  - Hidden files
  - Slack space
  - Swap files
  - Index.dat files
  - Hidden ADS
  - Windows Search index
  - Unallocated clusters
  - Unused partitions
  - Hidden partitions
  - Registry settings
  - Windows event logs
  - ...

Underlined: covered when we discussed file system forensics
Artifacts of user activities

- Volatile information
- Registry information
- More non-volatile information
System time & logged-on users

- System time

- Logged-on users
  - determine who is logged on to the system: locally or remotely

![Image of command output showing logged-on users]
If users logged into a system remotely, investigators should also see what files they have open, if any.
Netstat allows a user to collect information regarding network connections on a Windows system.
More volatile information

- **Process information**
  - Discover what processes are running on a potentially compromised system

- **Process-to-port mapping**
  - When there is a network connection open, find out which process is responsible for and using that connection

- **Network status**
  - Is the system connected or not? Collect NIC information

- **Clipboard contents**
  - Something a user copies to the clipboard on a Monday may still be there on Thursday

- **Command history**
  - To recover the command history (if an attacker clears the screen)
Artifacts of user activities

- Volatile information
- Registry information
- More non-volatile information
MRU lists

- MRU (‘most recently used’) lists contain entries made due to specific actions performed by the user.
- There’s numerous MRU lists located throughout various Registry keys.

**RunMRU:** When a user types a command into the 'Run' box via the Start menu, the entry is added to this Registry key.
MRU lists: Another example

- BagMRU: contains information of last visited folders
<table>
<thead>
<tr>
<th>MRU lists: Examples...</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XP Search Files</strong></td>
</tr>
<tr>
<td><strong>Internet Search Assistant</strong></td>
</tr>
<tr>
<td><strong>Printers, Computers and People</strong></td>
</tr>
<tr>
<td><strong>Pictures, music, and videos</strong></td>
</tr>
<tr>
<td><strong>XP Start Menu - Recent</strong></td>
</tr>
<tr>
<td><strong>R. Desktop - Connect</strong></td>
</tr>
<tr>
<td><strong>Run dialog box</strong></td>
</tr>
<tr>
<td><strong>Regedit - Last accessed key</strong></td>
</tr>
<tr>
<td><strong>Regedit - Favorites</strong></td>
</tr>
<tr>
<td><strong>MSPaint - Recent Files</strong></td>
</tr>
<tr>
<td><strong>Mapped Network Drives</strong></td>
</tr>
<tr>
<td><strong>Computer searched via Windows Explorer</strong></td>
</tr>
<tr>
<td><strong>WordPad - Recent Files</strong></td>
</tr>
<tr>
<td><strong>Common Dialog - Open</strong></td>
</tr>
<tr>
<td><strong>Common Dialog - Save As</strong></td>
</tr>
<tr>
<td><strong>WMP XP - Recent Files</strong></td>
</tr>
<tr>
<td><strong>WMP XP - Recent URLs</strong></td>
</tr>
<tr>
<td><strong>OE 6 Stationery list 1 - New Mail</strong></td>
</tr>
<tr>
<td><strong>OE 6 Stationery list 2 - New Mail</strong></td>
</tr>
<tr>
<td><strong>PowerPoint - Recent Files</strong></td>
</tr>
<tr>
<td><strong>Access - Filename MRU</strong></td>
</tr>
<tr>
<td><strong>FrontPage - Recent lists</strong></td>
</tr>
<tr>
<td><strong>Excel - Recent Files</strong></td>
</tr>
<tr>
<td><strong>Word - Recent Files</strong></td>
</tr>
</tbody>
</table>
UserAssist key: indicate last accessed system objects

- E.g., Control Panel applets, shortcut files, programs, etc.
- HCU\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist
We can gain a better understanding of what types of files or applications have been accessed on a particular system, e.g.:

- The decoded value shows a potential amount of information:
  - name of user profile - 'Cpt. Krunch' - from which the .exe was executed
  - researching 'p2ktools.exe', it is used for managing Motorola cell phones
  - user has p2ktools folder in parent directory called 'Razor programs'
    - tells both location and indicator that the suspect has a Motorola Razor cell phone
Autorun locations

- Registry keys that launch programs or apps during boot
  - E.g., in a system intrusion, autorun locations could reveal the installation of a trojan backdoor

- List of common autorun locations:

  - HKLM\Software\Microsoft\Windows\CurrentVersion\Runonce
  - HKLM\Software\Microsoft\Windows\CurrentVersion\policies\Explorer\Run
  - HKLM\Software\Microsoft\Windows\CurrentVersion\Run
  - HKCU\Software\Microsoft\Windows NT\CurrentVersion\Windows\Run
  - HKCU\Software\Microsoft\Windows\CurrentVersion\Run
  - HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
  - (ProfilePath)\Start Menu\Programs\Startup
Anytime a device is connected to the Universal Serial Bus (USB), drivers are queried and the device's information is stored into the Registry (i.e., thumb drives).
1. Department manager alleges that individual copied confidential information on DVD.
2. No DVD burner was issued or found.
3. Laptop was analyzed.
4. Found USB device entry in registry:
   PLEXTOR DVDR PX-708A
5. Found software key for Nero - Burning ROM in registry
6. Therefore, looked for and found Nero compilation files (.nrc). Found other compilation files, including ISO image files.
7. Image files contained DVD-format and AVI format versions of copyrighted movies.

**Conclusion:** No evidence that company information was burned to disk. However, laptop was used to burn copyrighted material and employee had lied.
Application-specific information

- E.g., the Internet Explorer uses the Registry a lot
  - Under the HKCU\Software\Microsoft\Internet Explorer key
From previous example, an examiner could conclude:

- the user possibly has a gmail and hotmail email address
- engages in online banking at tdbanknorth
- is interested in digital forensic websites
- perhaps go to college at Champlain
- has been researching apartments in the area
- …
More Registry artifacts

- **System information**
  - Computer name, OS version, last shutdown time

- **Time zone information**
  - Important for establishing a timeline of activity on the system

- **Wireless SSIDs**
  - List of service set identifiers (SSIDs) to which it has connected

- **Shares**
  - Remotely shared resources, e.g., disk volumes (can be hidden)

- **Audit policy**
  - Indicates types of events recorded in the Event Log

- **Mounted devices**
  - Information about devices and volumes mounted on NTFS

- **Users**
  - Account creation time, names, last login time, last failed login attempt, account expiration, etc.

- **...**
More Registry artifacts

- System information
  - Computer name, OS version, last shutdown time
- Audit policy
  - Indicates types of events recorded in the Event Log
- Time zone information
  - Important for establishing a timeline of activity on the system
- Wireless SSIDs
  - List of service set identifiers (SSIDs) to which it has connected
- Shares
  - Remotely shared resources, e.g., disk volumes (can be hidden)

Homework

Locate the Registry keys where this information can be found.
Artifacts of user activities

- Volatile information
- Registry potpourri
- More non-volatile information
The Recycle Bin allows user to retrieve and restore files that have been deleted.

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Common File System Structure</th>
<th>Location of Deleted Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 95/98/ME</td>
<td>FAT32</td>
<td>C:\Recycled\INFO2</td>
</tr>
<tr>
<td>Windows NT/2K/XP</td>
<td>NTFS</td>
<td>C:\Recycler\INFO2</td>
</tr>
<tr>
<td>Windows Vista</td>
<td>NTFS</td>
<td>C:$Recycle.Bin\</td>
</tr>
<tr>
<td>Windows 7</td>
<td>NTFS</td>
<td>C:$Recycle.Bin\</td>
</tr>
</tbody>
</table>

The user’s deleted file is placed within the file under a subdirectory named with the user’s security ID, e.g.,

C:\RECYCLER\S-1-5-21-1454471165-630328440-725345543-1003
Interesting files

- **Link files**
  - .LNK files: shortcuts that point to another file or folder
  - May indicate the user opened a file; contains file’s details

- **Prefetch files**
  - .PF files are a specialized file type, to speed up the running of programs
  - Contains info about the last time the program was executed

- **Installed programs**

- **Thumbnail cache files**
  - thumbs.db: thumb cache of files browsed in the Thumbnails view

- **Printer files**
  - Contain information about printing jobs

- **Pagefile.sys and Hiberfil.sys**
  - The swap file and the file for storing RAM contents upon hybernation
Windows is the most widely available operating system on desktop platforms.

Due to its central role in setting up and supervising the system, Windows maintains valuable data structures for forensic investigators: the Registry, and the Event Log.

By analyzing these and over volatile/non-volatile pieces of information, investigators can gather a wealth of information about user activities on the computer.
References

- Primary bibliography
  - [Carrier 2005], Chapter 17
Next class

- Server forensics: Linux