Windows Forensics Introduction
The Windows Operating System

- Currently the most common for caseload, with OSX / MacOS becoming more and more frequent.
- Big, complex behemoth of an OS.
- Does a lot of things under the hood.
- Does a lot for compatibility and “experience”.

Mountains & Minds
Windows Forensics Core

• Most time spent in Windows forensics understanding live artifacts if possible (running processes, network connections, etc.).

• If volatile data is not available – spending a lot of time digging into the operating system itself (typically in the registry) and available logs.
Executing Code on Windows

• Sounds straightforward. Isn’t.
• Portable executables (.exe), DLLs (.dll)
• Execute directly on the command line or through other binaries.
• Common paths to run malware as a .dll / library to decrease detection rates.
• Executable scripting languages (Jscript, VBScript, Powershell).
• Other stuff (HTA files, SCR files, etc.).
• Powershell.
Credentials on Windows

- NTLM and Kerberos are primary mechanisms.
- SSO Single Sign On credentials are cached by authentication providers on the system.
- Leads to pass-the-hash/credential attacks.
- Stored locally in the SAM Security Account Manager file
  - HKLM\SAM
  - HKLM\SYSTEM
Windows Persistence

• Many ways to perform this but the most frequent areas are referred to as “autoruns”.
• Windows registry is the most common place to find autoruns:
  — HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
  — HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run Once
  — HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
  — HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
• Installing / modifying a Windows Service is also a popular persistence mechanism.
Process / Command-line Logging

- Windows doesn’t have default process or command-line logging.
- Command-line logging may be configured locally
  - AKA Event 4688
  - AKA “Command line process auditing”
- Most often configured in:
  - Endpoint Detection and Response (EDR) tools
  - Microsoft’s SYSMON
Powershell Logging

• Since Windows 7 – primarily Powershell v5
• Default logging (mostly) disabled, however, logging for anything that the Antimalware Scan Interface (AMSI) determines to be “suspicious”.
• Events Logged as a 4104 event – level “Warning”
• Better logging in Powershell is configured through a Group Policy Object (GPO)
  — (Administrative Templates -> Windows Components -> Windows PowerShell)
  — Can configure module, script block and transcription logging
Powershell Logging

• Module Logging
  — EID 4103
  — Captures portions / snippets
  — Does not reliably capture commands executed

• Script Block Logging
  — EID 4104
  — Show all commands and / or source for any PowerShell run on the system, user, and path to the script

• Transcription Logs
  — Keystroke / over-the-shoulder logging of a PowerShell session
Powershell Logging

• Powershell Tricks...
  ‒ There are actually a lot of methods for obfuscating and executing commands in PowerShell and we see them often in incident response and forensics.

• All of these are the same:
  ‒ Fully spelled out:
    • powershell.exe –EncodedCommand ZQBjAGgAbwAgAClARAbvAHIAbwB0AGgAeQAiAA==
  ‒ Truncated with alternate capitalization:
    • powershell.exe –eNco ZQBjAGgAbwAgAClAVwBpAHoAYQByAGQAIgA=
  ‒ Using caret escape-character injection to break-up the string:
    • powershell.exe –^e^C^ ZQBjAGgAbwAgAClAVwBpAHQAYwBoACIA
CyberChef

• At some point we have to discuss CyberChef – we’ll start here and then discuss further in network forensics.
• CyberChef:
  — Can host locally or access online - [https://gchq.github.io/CyberChef/](https://gchq.github.io/CyberChef/)
  — Data swiss army knife
  — We’re going to use for encoded PS> commands
Into the CyberChef!
Windows Registry Forensics
Registry Refresher

• Basically a filesystem into and of itself.
• Hierarchical database that primarily stores information as a key->value pair.
• Core store for compatibility and ease-of-use / preferences features in Windows, also used by installed software.
• Registry can be obtained via disk capture OR memory capture.
Registry Hierarchy

- **HKEY_CURRENT_USER (HKCU)**
  - Configuration information for currently user (associated with their profile)
- **HKEY_USERS (HKU)**
  - Local profiles on the system.
- **HKEY_LOCAL_MACHINE (HKLM)**
  - System-level configuration information
- **HKEY_CURRENT_CONFIG**
  - Contains information about the hardware profile of the system
Registry Key Forensic Features

• This section will be a review / survey of key locations in registry that store forensically interesting and important information.

• This typically involves when the subject of an investigation takes actions to:
  — Add registry keys (e.g., persistence).
  — Modify registry keys (running an executable)
SHIMCACHE

- Application Compatibility Database
- WinXP+
- \SYSTEM\CurrentControlSet\Control\Session Manager\ AppCompatCache
- Entry for every application executed:
  - Full path information.
  - Last modification time.
  - File size.
  - Execution flag.
- Notes
  - Sequential (typically in order of execution)
  - Has a max # of entries (usually 1,024)
USERASSIST

Windows Explorer UserAssist

- Application Compatibility Database
- WinXP+
  - NTUSER.DAT\Software\Microsoft\Windows\Currentversion\Explorer\UserAssist\{GUID}\Count
  - Tracks GUI executables, shortcuts execution

- Notes
  - All values encoded
  - Uses GUIDs
    -  CEBFF5CD – Executable file
    -  F4E57C4B – LNK / Shortcut
  - Good for seeing what happened in a user session.
OpenSave MRU

OpenSave – User / Session Feature

• Registry key
• WinXP+

  — NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSavePIDIMRU
  — Tracks files that have been opened / saved through the Windows Explorer shell dialog box
  — Organized into 2 subkeys “*” and the file extension subkeys

• Entry includes:
  — File Name
  — Full Path
  — MRU order (order in which file was opened/saved by user)

• Notes
  — Good for seeing what happened in a GUI user session.
  — Not super helpful for malware investigations.
SRUM

System Resource Usage Monitor (SRUM)
- Records programs and network activity
- Win8+
- Entry includes:
  - Timestamp
  - EXE path
  - SID
  - BytesRecieved / BytesSent
- Notes
  - Writes once and hour and at shutdown.
  - Can grab from disk at C:\Windows\System32\sru\SRUDB.dat
BAM!

Windows Explorer UserAssist
• Background Activity Moderator (BAM)
• Win10 1709+
  — HKLM\SYSTEM\CurrentControlSet\Services\bam\UserSettings\{SID}
• Entry includes:
  — Last execution timestamp
  — EXE path
• Notes
  — Only exists in recent Win10
Recent Office Files

- The Office apps trace recently executed / edited files
- Office 2010+
  - NTUSER.DAT\Software\Microsoft\Office\<Version>15
  - NTUSER.DAT\Software\Microsoft\Office\VERSION\UserMRU\LiveID_####\FileMRU
  - Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs
- Notes
  - Variety of values here
  - For RecentDocs – will need to do some decoding (hex)
  - Organized by file extension