WFA: Locating Wireless SSIDs within Windows images

Windows systems record a great deal of user activity, under the guise of optimizing the "user experience" (Note: Windows XP gets its name from the "user eXPerience"). This information is recorded and maintained automatically, without any user interaction (though some specifics steps may be taken to prevent the data from being recorded) and can be extremely valuable to a forensic analyst.

Whenever a user connects to a wireless service set identifier (SSID[1]), the software that manages the wireless network connections may record that information within the Registry or file system for later use. On many Windows systems, this may be the Wireless Zero Configuration Service (WZCSVC); on other systems, this may be software specific to the wireless interface driver or vendor. These software products may maintain a listing of SSIDs connected to, and this information may be valuable to a forensic analyst in cases in which corporate data has been copied to a laptop, and then the user has taken that laptop to a local wireless hotspot and sent that information to a competitor via a web-based email program. Another example where such information may be useful is where harassing emails have been received, and the IP addresses in the email headers traced back to popular wireless hotspots (i.e., Starbucks, Kinkos, etc.)…an examination of the suspect's system may reveal that they'd connected to an SSID associated with that hotspot.

SSIDs in the Registry

[1] http://en.wikipedia.org/wiki/SSID

If WZCSVC is used to manage the wireless connections on a system, SSIDs are maintained in the following Registry key:

HKLM\SOFTWARE\Microsoft\WZCSVC\
Parameters\Interfaces

Beneath this key, you may see at least one subkey, which looks like a globally unique identifier (GUID) which refers to a specific interface.

A list of network interfaces used on a system is maintained in the following Registry key:

HKLM\SOFTWARE\Microsoft\
Windows NT\CurrentVersion\
NetworkCards

The subkeys beneath the NetworkCards key are numbered, and each contains a "Description" and "ServiceName" value. The "Description" value contains a friendly name for the interface, and the "ServiceName" value contains the GUID for the interface. The GUID can then be correlated to the contents of the following Registry key to obtain specific settings such as if DHCP is enabled, the IP address assigned to the interface, etc:

HKLM\SYSTEM\ControlSet00*n*\Services\
Tcpip\Parameters\Interfaces\{*GUID*}

Within the GUID subkey, you will see values that start with the word "Static#" and are appended with four digits, such as "0000", "0001", "0002", etc. One or more of these values may exist. These values are binary data types, and contain the SSID name and the date that the SSID was connected to within that binary data. The SSID name is located at an offset 20 bytes

into the binary data and is a null-terminated string (that may be up to 16 bytes), as illustrated in figure 1.

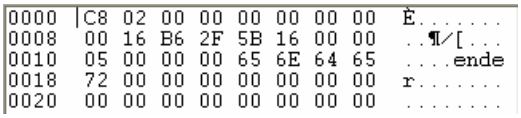

Figure 1

The Software Registry hive file can be extracted from the image, and the Parse::Win32Registry Perl module can be used to extract the binary data for each value from the Registry key, and parse the name from within that data using Perl code such as the following:

```
my $name = substr($data,0x14,
  0x10);
$name =~ s/\00//g;
```

The date that the SSID was connected to is an 8-byte FILETIME[2] object located beginning at an offset 696 bytes into the binary data, as illustrated in figure 2.

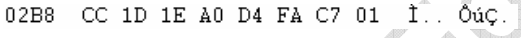

Figure 2

This FILETIME object is easily converted into a 32-bit Unix time value and processed using the Perl `gmtime()` function to display the time in Universal Coordinated Time (UTC) format.

> Windows uses FILETIME objects to represent timestamps in other locations throughout the Registry and file system, to include generated and written times for Event Log records (specifically on Windows 2000, XP, and 2003). You must incorporate the ActiveTimeBias value from the TimeZoneInformation Registry key in order to display the timestamp as it was seen by the user on the system.

[2] http://support.microsoft.com/kb/188768

Other SSID Management Applications

The MS WZCSVC is not always the application used to manage wireless connections and maintain SSID information. For example, some Dell systems use the Broadcom drivers for NICs, and information about SSIDs appear to be maintained in the following Registry key:

HKLM\SOFTWARE\Broadcom\802.11\
{*GUID*}

Values of interest within this key include "Static#000*n*" (which contains binary data that appears to be obfuscated in some manner) and "TimeStamp" (a 32-bit Unix time value).

In one image of a Lenovo ThinkPad system running Windows XP, the following Registry key was found to contain information about SSIDs:

HKLM\Software\Lenovo\
Access Connections\Locations*SSID*\

Each of the *SSID* subkeys were the names of the SSIDs that the system had been connected to, and the Registry key LastWrite times (also FILETIME objects) may have indicated times for when the system had been attached to the SSID. Further information can be found beneath this key, as well. For example, the following key contains values that will tell you if DHCP and the firewall were enabled:

HKLM\Software\Lenovo\
Access Connections\Locations*SSID*\
AdptList\Adpt00

Articles located in online forums indicate that a value named "m_szHomePage" may be found within this Registry key, and be used to specify the location of the Internet Explorer homepage.

Additional information about the SSID itself may be located in the following Registry key path:

HKLM\Software\Lenovo\
Access Connections\Locations*SSID*\
AdptList\Adpt00\SsidList\Ssid00

Systems using the Intel PROSet/Wireless drivers (some Dell systems, for example) may have information about the software located in the following Registry key:

HKLM\SOFTWARE\Intel\Wireless

Summary

Windows systems maintain a great deal of system- and user-specific information that may be valuable to a forensic analyst when pursuing an investigation. The location or value of the information may not be immediately obvious, and may require some manual processing (i.e., extract the Registry hive files from the image and load those hives into RegEdit) as well as exploration via vendor support sites.

Checks for these Registry keys should not be dismissed for desktop systems. Many modern desktop systems come with built in wireless capability, and wireless access points within range of the interface can be connected to and used to access the Internet. Once Registry keys that illustrate connectivity to SSIDs have been located, the pertinent information (i.e., Registry key LastWrite times, value names, specific information parsed from within binary values, etc.) can be programmatically retrieved and displayed. Manual searching of the Registry hive files can be performed by extracting the hive files from the image and loading the hive file into RegEdit.

Resources

- Lance Mueller's EnScripts[3] for extracting wireless SSIDs from a Windows image

[3] http://www.forensickb.com/2007/08/enscript-to-list-wireless-ssids.html

Harlan is a forensic analyst located in the metro DC area, and is the author of three books related to live response and forensic analysis of Windows systems. He also presents at conferences and provides training on those same topics, as well. Comments and questions can be directed to him at keydet89@yahoo.com.