Windows Services: Section 1 Transcript

Introduction

Welcome to the Windows Services module.

During this module, we'll introduce several items and definitions related to command syntax, CLI tools and system characteristics.

Throughout this module, you'll be presented with the opportunity to assess and apply what you've learned.

At the end of this module, you will be able to:

- Perform a system characterization for Windows XP/2003 based on the information presented,
- Perform a system characterization for Windows Vista/7/2008 based on information presented,
- Manage system services using Windows (CLI) tools, and
- Manage system services using Sysinternal tools.

Bypass Exam Introduction

If you are already familiar with the subject matter presented in this module, you can choose to take a Bypass Exam to skip this module.

The Bypass Exam option provides a single opportunity to successfully demonstrate your competence with the material presented within the module. If you pass, you'll receive credit for completing the module, unlocking the content within, and you will be free to proceed to the next module. If you do not pass, you will need to successfully complete the module, including all exercises and the Module Exam, to receive credit.

Click the Next Section button to continue.

Windows Services: Section 2 Transcript

Device Drivers

Device drivers are kernel-mode processes that Windows uses to interact with hardware devices. The I/O Manager calls functions in the device driver to get the mouse, network card, etc. to carry out the actions to perform its job.

There are three possible outcomes for the I/O request:

- Queue for later process,
- Send to hardware port, or
- Send to another device driver.

Many device drivers are loaded and configured automatically by Plug and Play. They also interface with the Power Manager and Device Manager. The Power Manager handles all power settings while the Device Manager stores configuration information for the devices. For example, the Hardware Abstraction Layer (HAL.dll) stores platform specific functions and hardware details.

Device Drivers

Windows automatically configures a device so that it works properly with other devices that are installed on the computer. As part of the configuration process, Windows assigns a unique set of system resources to each device so that it can function with little or no user input.

Windows device drivers included on the Setup CD are stored in a single cabinet file, Driver.cab. This file is used by Setup and other system components as a driver file source. Click each component to learn about them.

Win32 API
32-bit application programming interface for Windows operating systems that initiates I/O operations through Ntdll

Ntdll
Function library that exposes stubs that invoke Windows system calls (like NtReadFile below) as exported functions

NtReadFile
I/O Manager system call that creates an I/O request packet and directs it to the appropriate driver

I/O Manager
Subsystem that controls and interacts with all devices on the system and provides routines that drivers can call to have the I/O Manager insert IRPs into the associated device queue

IRP
I/O Request Packet sent by I/O Manager to request services from a driver

IoCallDriver
Routine that sends an IRP to the driver associated with a specified device object

Device Drivers 3/5

Information files (.inf files) are searched when Windows starts or new hardware is detected. These text files provide the names and locations of driver-related files and the initial settings required for new devices to work. During setup, Driver.cab is copied from the installation CD to the WINNT\Driver Cache\platform directory of the local hard disk. The variable platform here is a placeholder for a specific value that reflects the architecture of the system: for example, i386.

Device drivers are installed with no user intervention if certain conditions are met:

- Installing the driver does not require showing a user interface.
- The driver package contains all files needed to complete the installation.
- The driver package is available on the system in the Driver.cab file or was previously installed.
- The driver package is digitally signed.
- No errors occur during installation.

If any condition is not met, the device installation restarts and the user might need to respond to dialog boxes or messages.

Device Drivers 4/5

Manual installation of a driver requires that the person installing the driver be a member of the Administrators group. Windows determines which device driver to load for a device using the following criteria.

- Driver ranking schemes
- Driver search location policies
- Windows Driver Protection
- Windows Update

To facilitate device installation, devices that are set up and configured in the same way are grouped into a device setup class. For example, SCSI media changer devices are grouped into the MediumChanger device setup class.

Section Completed 5/5

Windows Services: Section 3 Transcript

Services

Windows Services are long running applications that run in their own Windows sessions. Each service provides a specific functionality to the operating system (OS). Services can be started manually or automatically when the system boots up, or they can be paused, restarted, or stopped. Service programs do not display any user interface.

Services generally run in the security context of the system account or the Service Control Manager (SCM), but they can also be run in the security context of a specific user. The configuration information for services is stored in subkeys under the **registry** location HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services.

Service Control Manager

Service applications conform to the interface rules of the Service Control Manager (SCM). Through configuration settings, services can be configured to start automatically during system startup, when a user logs on to the system, manually, or by an application that uses the service. A driver service conforms to the device driver protocols. They are similar to service applications, but they do not interact with the SCM.

SCM Functions

Here are a few programs that use SCM functions: Service Program, Service Configuration Program, and Service Control Program (SCP).

- The Service Program contains executable code for one or more services, and can be configured to execute in the context of a user account from either a local, primary, or trusted domain,
- The Service Configuration Program queries or modifies the services database, and
- The SCP starts and controls services.

These programs are important to understand because services need to be understood in the context of the system environment.

Service Start

How do you know when a service starts? Well, in HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\[service name], each service is assigned a start value.

There are five values as shown in the chart.

- The start value of 0x0 (BOOT_START) means the service is started at boot by ntldr or

winload.exe.
- The start value of 0x1 (SYSTEM_START) means the service is started when the system loads by ntoskrnl.
- The start value of 0x2 (AUTO_START) means the service is started automatically by the SCM.
- The start value of 0x3 (DEMAND_START) means the service is started manually on demand. And,
- The start value of 0x4 (DISABLED) means the service is disabled.

Load Order 5/10

Now that you understand that many services start automatically, how does the system know what order to load them? It's actually pretty simple. It's determined by:

- Order of Services within a group (ServiceGroupOrder),
- Load Ordering Group list (GroupOrderList), and
- Dependencies listed under each service.

The values ServiceGroupOrder and GroupOrderList are located at HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control.

Windows Service Hardening (WSH) in XP 6/10

Similar to how Windows checks driver signatures for security reasons, services are executed in the security context of a user account. There are a few predefined special accounts:

- LocalSystem, Security Identifier (SID) S-1-5-18, has extensive local privileges and acts as the computer on the network.
- LocalService, SID S-1-5-19, has minimum local privileges and has anonymous credentials on the network.
- NetworkService, SID S-1-5-20, has minimum local privileges and acts as the computer on the network.

If these do not meet your requirements, then you can specify a user account.

Windows Service Hardening (WSH) in Vista Plus 7/10

Windows services have been a favorite target of malware and hackers because many services start automatically, run continuously, communicate with the network, and have a high level of privilege. Malicious software targets services that can be piggybacked in order to exploit these services and enter the system.

Windows Vista, and later OS versions, use several privilege restrictions and identifiers to reduce the chances that an attacker who manages to compromise a service will actually be able to do damage to the system. Windows Service Hardening restricts critical Windows Services by ensuring that the services run under the least privileged accounts necessary for the job being performed. Click each of the Windows Service Hardening Components to find out more about them.

Why Windows Service Hardening (WSH)
Service containment

- Prevents code injection
- Prevents services from altering the configuration of other services

Service Hardening
Specific to Vista+

- Service Resource Isolation
- Least Privilege Services
- Restricted Network Access
- Session 0 Isolation
- Kernel Mode Code Signing (KMCS)

Service Resource Isolation
Service specific SIDS

- Primary and Secondary SIDS
- Secondary SIDS are S-1-5-80-{SHA1 hash of short svcname}
- ACLs on files and registry keys can be locked down to only the SID attributed to the service
- Restricted SIDS
- Write restricted SIDs added
- Now both SIDs are checked, Service SID and Restricted SID for access
- The combination prevents access to other services if they are not explicitly granted access

Least Privilege Services
Services are now able to provide the SCM(service control manager) with a list of required privileges

- SCM uses LSA to remove permissions not required
- Runs services under lower-privileged accounts
- Moved eight services out of SYSTEM and into LocalService
- Moved Four services to NetworkService
- Six new svchosts have been implemented

Restricted Network Access
Firewall Policies can now be applied to individual services
Filtering Capabilities:

- Direction
- Protocol
- Principal
- Interface

Session 0 Isolation
Prevents shattering attack (privilege-escalation)

- Previous versions of Windows users logged on with session ID of zero
- This session was shared with services
- Now when users log in the session ID is incremented
- This prevents users from being able to send messages to services
- Session 0 does not allow interactive

WSH Vista + & KMCS

On 64-bit versions of Windows Vista and later, Kernel Mode Code Signing (KMCS) will only load a kernel-mode driver if it has a digital signature. In 32 and 64-bit versions of Windows, boot-start drivers must contain an embedded signature. That's where signature checking comes into play. Signature checking verifies if a driver should be loaded, or if it's a security and/or stability risk.

Windows has three driver signature checking levels when installing a driver:
At level 0, the system silently checks and logs unsigned drivers, and allows their installation. At level 1, the system displays a warning message. The driver can be installed by the user, but it is not recommended. At level 2, the user is notified that the driver installation is blocked.

The driver signature checking level is defined in the registry in the policy subkey located at HKEY_LOCAL_MACHINE\Software\Microsoft\Driver Signing.

CLI Tools

To learn about various command-line interface (CLI) tools, click the on-screen CLI tools to learn about them.

Services Controller (SC): The SC tool is an interface that communicates with the Windows Services Controller and the installed services. This tool can be used to start and stop services, as well as view and modify the properties and configuration of a service.

Net (Start/Stop): Although the net command has a number of sub-functions, the start and stop commands can be used to start and stop specific services on the system. These commands do not provide any additional functionality to interact with the computer's services.

Drivers: This tool is part of the Windows Resource Kit tools for performing administrative tasks and allows you to list the drivers that are installed on the system and view the characteristics associated with the drivers.

Driverquery: This tool is native to Windows and comes as part of the internal commands included with cmd.exe. It performs the same function as drivers.exe, providing a list of the installed drivers and properties associated with each.

Sigcheck.exe: Sigcheck is a command-line utility that shows file version number, timestamp information, and digital signature details, including certificate chains. It also includes an option to check a file's status on VirusTotal, a site that performs automated file scanning against over 40 antivirus engines, and an option to upload a file for scanning.

Signtool.exe: Sign Tool is a command-line tool that digitally signs files, verifies signatures in files, and time-stamps files. This tool is automatically installed with Visual Studio. To run the tool, use the Developer Command Prompt (or the Visual Studio Command Prompt in Windows 7). For more information, see Visual Studio Command Prompt.

Exercise Introduction

It is time for an Exercise. In order to successfully complete the Exercise, you are expected to use your notes and any available resources presented throughout the course, in addition to conducting your own Internet research.

Windows Services: Section 4 Transcript

System Characterization (Process List)

In this section, we will discuss how to perform a passive characterization of a Windows system using only the information provided in a process list. Although this type of characterization is not all-inclusive, and does not provide definitive answers when characterizing the OS, it is a starting point that enables narrowing the scope of possibilities when describing the OS's version and possible role.

OS Version

In the sample process list, the first item to examine is the PID of the System process. The value of the System PID narrows down which version of the OS you are looking at. A PID of two indicates it is a Windows NT system, a PID of eight indicates it's Windows 2000, and a PID of four indicates it's Windows XP or newer.

Other items to consider when characterizing the OS include other processes that are running:

- MsMpEng is a component of Windows Defender and provides anti-spyware services.
- NisSrv is a part of Microsoft Security Essentials. It is the Microsoft Network Real-Time Inspection Service.
- msseces is a part of Microsoft Security Essentials. It is the client interface to MSE.
- wininit was introduced with Windows Vista.
- csrss is a user-mode portion of the Windows subsystem. In Windows Vista+, two or more instances of this process are visible. If only one instance of the process is present, the OS version is no higher than Windows XP/2003.
- dwm is the Desktop Window Manager, first introduced with Windows Vista, which provides the transparency effects for the Windows and dialog boxes on the desktop.

System Up-time

The best way to learn the system up time is to look at the smss elapsed time. It provides the closest to system up time for the target system.

Hardware Configuration

The on-screen example can also tell us a little bit about the hardware configuration of the target system.

The number of CPU cores available to the computer's operating system can be determined by comparing the smss elapsed time and the Idle CPU time. The Idle thread count also depicts the number of CPU cores available to the system. Idle appears to have two threads and the Idle CPU time is approximately double the total up time; therefore, it's a dual processor (each single core) or a single, dual-core processor machine.

In addition to learning that this is a dual processor machine, we can look at a few other items. The audio process and video process listed are not associated with high-end audio and video processing. It's also a safe assumption that this computer is hosting virtual machines since we see several VMware processes running that are associated with hosting Virtual Guest images.

User logon Time

The Elapsed Time of the shell process tells us how long a user session has been active. Since we know that explorer.exe is the default GUI shell, we can assume that a user has been logged on for the amount of time depicted in the Elapsed Time column.

In this example, it can be determined that a user is logged on for approximately two hours, 25 minutes, and 58 seconds.

Role of the Box

We can determine if the computer is a workstation or a server by looking at the process list. Client processes identify a system as a Windows workstation, and are usually found above user processes, starting with explorer. Server processes identify a system as a Windows server.

The absence of any server-specific programs in the provided example is an indication that this computer is likely a workstation.

Nature of the User

Evaluating the nature of the user requires an examination of the applications the user is running, and an understanding of the types of actions that can be performed using those applications.

For example, a user that is running web browsers, email applications, or those associated with social media or entertainment is considered a typical home user. A user that is running a command shell, a kernel debug program (such as KdLive), or a network traffic analysis application (such as WireShark) is considered an advanced user that probably understands how computers work and how networks function.

This evaluation is not an exact science; it's an educated guess; however, it is valuable to determine the technical competency of the target system's user. This can be used to evaluate the risk of being discovered during an operation, and the risk of alerting the user that something or someone is on their machine.

In the example, we can determine that this user is a general home user, because there's no evidence of programs being used to perform analysis or other advanced tasks.

Suspicious Processes

When determining if there is suspicious software running on a system, look for processes that appear to be out of the ordinary, those with strangely spelled process names, or those that are using an excessive amount of resources. If you identify a process as suspicious, verify that the process is

malicious or trustworthy through further analysis.

As you can see, there are three instances of Microsoft Internet Explorer running. Note the number of resources being used by the third instance; the number of threads, handles, and private address space is much greater than the other two instances. This should raise a red flag, that this process is possibly malicious. In this case, after further investigation, we discovered that this instance of Internet Explorer is associated with Windows Update, and is a legitimate process.

Exercise Introduction

It is time for an Exercise. In order to successfully complete the Exercise, you are expected to use your notes and any available resources presented throughout the course, in addition to conducting your own Internet research.

Windows Services: Section 5 Transcript

Summary

You have completed the Windows Services module.

This module included system characterization for Windows XP/203 and Vista/7/208, Windows CLI tools and Sysinternals tools. You should now be able to:

- Perform a system characterization for Windows XP/2003 based on the information presented,
- Perform a system characterization for Windows Vista/7/2008 based on information presented,
- Manage system services using Windows (CLI) tools, and
- Manage system services using Sysinternal tools.

To receive credit and advance to the next module, you must achieve a passing score on the Module Exam. Click the Next Section button to begin the Module Exam.