CHAPTER EIGHT

Systems Software and IT Operations General Controls

VIRTUALLY EVERY COMPUTER SYSTEM has some type of master program—often called an operating system (OS)—to perform such tasks as scheduling an application program to run or saving or storing the results of an application program and outputting the results to a printer or display device. No matter what the size of the computer operations, these OS master programs and related supporting software control a computer system's operations. Chapter 6 discussed the importance of information technology (IT) general controls, the kinds of controls covering all aspects of IT operations, and Chapter 7 discussed IT infrastructure controls, processes to better manage and control IT operations. This chapter discusses the very important area of systems software and related IT operations general controls.

Whether it is a Microsoft Windows operating system on a laptop computer or Linux controlling an office server system, the OS is a key component to any computer system operation. It and the supporting systems software are essential for supporting enterprise IT operations. IT auditors need to understand the importance of the OS as well as any operations control programs in the enterprises where they are performing general controls reviews.

This chapter discusses OS concepts and surveys some of the OS types and the supporting systems software that are essential in today's typical IT operation. IT auditors should have a very general understanding of the purposes and importance of these types of IT software and should look for effective general controls when performing reviews in this important general controls area. This chapter focuses on some of the more common types of operating systems found in today's IT systems as well as the supporting software to create well-controlled IT systems operations.

IT OPERATING SYSTEM FUNDAMENTALS

An IT or computer OS is the master program that serves as the prime interface between all system hardware components and the users of that computer system—the people and sometimes other automated devices. An OS is responsible for the management and coordination of the many activities necessary to run some applications on a computer system. Many IT auditors first encounter an OS while using their laptop or desktop personal computers equipped with either Microsoft or Apple Macintosh OS programs. A version of Microsoft's Windows OS is perhaps the most familiar to many. In its multiple versions, it can be both a major help and sometimes a source of frustration.

The origins of today's OS go back to the days of mainframe computers, when the manufacturers of these devices (e.g., Burroughs, Control Data, Honeywell, IBM, Univac, etc.) each developed complex OS programs for their systems. Each was fairly unique to that computer manufacturer and its individual machine models. As an example, Univac, which is considered the founder of large-scale computer systems, had its model 1100 and 494 series of mainframe computers in the 1970s. Each Univac product line had its own OS that did not communicate with either those of other Univac machines and certainly not with those of competitors. In those early days of mainframe computers, many computer models were unique, but many of these machine OS controllers had innovative features. As example, the Univac 1100 series of computers had facilities for real-time processing in the 1970s, a feature that IBM, then the dominant computer manufacturer, did not introduce until the 1980s.

The concept of a mainframe OS really changed, however, when IBM introduced its System/360 series of computer and mainframe operating systems in the mid-1960s. The IBM System/360 was a family of mainframe computers available in widely differing capacities and price points and with a single OS, called OS/360, planned for every model. This concept of a single OS spanning an entire product line was crucial for the success of IBMs System/360. IBM's current mainframe operating systems, later called MVS and now z/OS, are distant descendants of this original system; applications written for the OS/360 can still be run on modern IBM machines.

Exhibit 8.1 illustrates how an OS interacts with both computer resources and its applications. In the center, the computer system primarily consists of its central processing unit (CPU) and random access memory (RAM) as well as the external storage devices. The CPU is the set of highly integrated circuits that define the logic of how the computer will operate. The OS gives instructions to the CPU and manages the other computer resources. The next ring out is the key software components, such as an Oracle database system. The applications are found at the outer ring. Users and IT auditors primarily deal with these outer ring applications.

Any modern OS, no matter the size of the computer system hardware, is a very complex set of software that goes beyond the skills of even the most expert systems programmers. While IT auditors regularly perform internal controls reviews of applications and review general control reviews of many aspects of the supporting control programs such as database processors, but too often the OS is taken as a given and ignored from an internal controls perspective. IT auditors should have a general

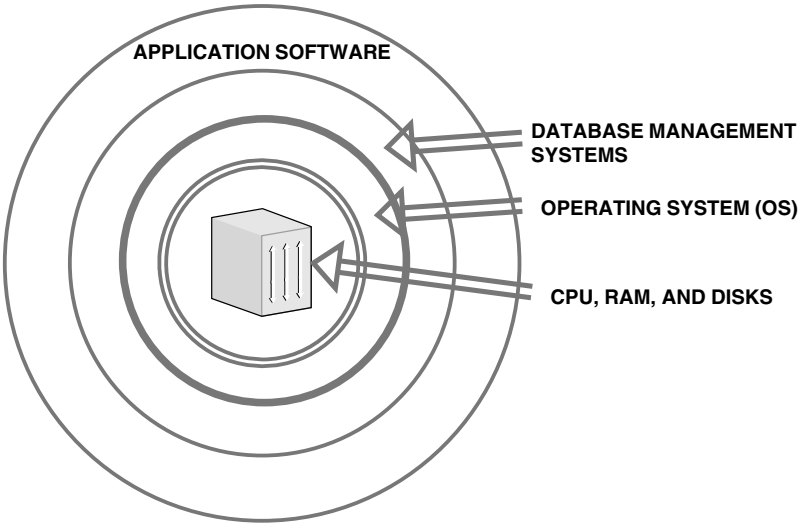

EXHIBIT 8.1 Role of the Computer Operating System

understanding of basic OS attributes and should consider reviewing general software and infrastructure controls surrounding an enterprise's computer systems OS resources.

Microcomputer or Personal Computer Operating Systems

The first microcomputers or personal computers (PCs) did not have the capacity or need for the elaborate operating systems that had been developed for mainframes or even what were called minicomputers; minimal operating systems were developed, often loaded from the computer's very limited read-only memory (ROM). An early disk-based operating system called Control Program for Microcomputers (CP/M) was found on many of the first early microcomputers. CP/M was popular on many of the first systems and was closely imitated by Microsoft's disk operating system (MS-DOS), which became wildly popular as the OS chosen for the IBM PC. IBM's version of that microcomputer OS was distributed by Microsoft in its early days and called IBM DOS. Microsoft began upgrading and expanding this OS through the stream of improved versions that are found on many laptop and desktop computers today, such as Windows XP or Windows 7.

Even before the IBM PC, Apple Computer Inc. (now Apple Inc.) launched a very popular microcomputer called the Apple II. A very popular machine then, the Apple II had its own primitive OS. After the introduction of the IBM PC, Apple soon launched a computer called the LISA, which had a very innovative graphical user interface (GUI) OS. LISA led to the Apple Macintosh computer with its Mac OS. Today, virtually every microcomputer OS today is based on either a Microsoft version of Windows or an Apple Mac OS.

Today's Server Operating Systems: Unix and Linux

The early mainframe computers in the 1970s were massive machines requiring complex OS supports and many environmental requirements. They were just too

complex for smaller businesses or university research facilities. As the computer industry grew during that decade, several manufacturers introduced smaller, less complex computers called minicomputers. Although no longer in existence today, the prominent computer manufacturers then were Digital Equipment Corporation (DEC) and Data General (DG). Each of these had lines of computer hardware started with their own OS facilities that were far less complex than the requirements of IBM's mainframe OS.

These minicomputers were developed before the Internet, as we know it, but there was a massive professional and academic interest at that time on finding better ways to improve the use computer resources. The Unix OS was an important product of this era. Unix (or UNIX) is an OS that originated at Bell Labs in 1969 as an interactive time-sharing system, and it has evolved as a nonvendor freeware product, with many extensions and new ideas provided in a variety of versions of Unix by different companies, universities, and individuals.

Because it was not a proprietary OS owned by any computer company and because it is written in the then-standard C programming language, Unix became the first open OS that could be improved or enhanced by anyone. A wide range of Unix versions were developed and used in workstation products from Sun Microsystems, Hewlett-Packard (now HP), IBM, and other companies. The Unix environment and its client-server program model were important elements in the development of the Internet and the reshaping of computing as centered on networks rather than in individual computers.

Unix has been designed to give software developers a single set of what are called application program interfaces (APIs) to be supported by every Unix system. An API is a written contract between systems and application developers defining what the two sets of developers are to receive and responsible for providing.

Unix has continued to be an important component of many IT systems. An IT auditor usually will have no need to review the ''innards'' of any Unix implementation, but he or she should ask some general questions about the status of any Unix system as part of any general controls review. Exhibit 8.2 contains some Unix IT general controls review questions. An IT auditor should look for implementation of one of the standard versions of Unix. Use of a standard version will allow an enterprise to better adapt and implement new applications going forward. That is, even if an enterprise is experiencing no control problems with applications currently running under a nonstandard version of Unix, problems going forward may arise if the installed versions of Unix is not compatible.

Although Unix had its origins as a free software facility where all interested users could upload versions and modify them for their own requirements, manufacturers gained the rights to Unix and established controls that limited the ability to change versions of the OS software. IT auditors generally think of a controlled software OS, such as Unix, as a good thing, but its many users in facilities, such as university research labs, were looking for a more open version of Unix that they could improve and tweak.

As an alternative to Unix, in 1991, Linus Torvalds, a programmer based in the Helsinki, Finland, developed a new OS, called Linux. Although similar to Unix, Linux had many features that impressed computer science professionals. The Linux OS is predominantly known for its use in servers, and it can be installed on a wide variety of

1. Meet with IT personnel to develop a general understanding of type and version of the installed Unix operating system (OS).
   a. Is the OS commercial software, such as Solaris or IBM's AIX?
   b. If the OS is freeware, such as from an academic source, determine that the version installed is current.
   c. Determine whether the installed OS is supported by appropriate documentation and backup procedures.
2. Obtain auditor-level permission to access and audit UNIX procedures. If not familiar with basic commands, request IT management to supply a tutorial.
   a. Obtain a login for auditor use.
   b. Create a special directory for auditor use, and change the permissions on this directory so that only the allowed IT auditor has access to this directory and the files within it.
3. Particularly if a smaller installation system, review the server's physical security to ensure that a casual outside user cannot, at a glance, see the entire setup of the server room. Ensure that the monitor is well hidden from open viewers.
   a. Review controls to ensure that only authorized persons can access the server.
   b. Determine that the server's CPU unit has adequate security to prevent unauthorized users from resetting it or even switching it off and that there are prevention mechanisms to protect against any booting from alternate media.
   c. Determine that only appropriate personnel are allowed physical access to the room through the use of locks or swipe cards.
4. Use what are called the Unix `eeprom` parameter to ensure that the UNIX *security mode* is set to ''full'' and that an adequate EEPROM password is chosen to ensure that even if a malicious attacker gains physical access to the system, he or she will be unable to reboot the system without knowing the EEPROM password.
5. Obtain the hardware inventory on the system, listing all hardware devices by issuing the UNIX command `usr/platform/'uname -i'/sbin/prtdiag -v`.
6. Get a list of software installed on the system through the UNIX command `pkginfo` to gather the details of the installed software, and resolve any differences.
7. What procedure is followed when adding new users? Ensure that there exists a well-documented procedure for adding new users.
8. Assess the adequacy of procedures for adding new users to the system, and review IT and HR records for the names and designation of these personnel.
   a. Determine that users have to fill in a form or sign their acceptance of an agreement before being given a username and a password and that users are required to sign their agreement and acceptance.
   b. Determine what password restrictions, consistent with enterprise security policies, are imposed on Unix server users.

EXHIBIT 8.2 Unix General Controls Review Procedures

computer hardware, ranging from embedded devices, mobile phones, and even some watches and supercomputers. The Linux OS, installed on both desktop and laptop computers, has become increasingly popular in recent years.

The feature that makes Linux unique is that it is totally ''freeware.'' That is, interested users can download a version of Linux at no cost, but they must agree to post any changes that they make to the software for all to see and use. All accepted changes become part of an enhanced Linux system. As an example of its growing acceptance, several years ago, even IBM installed Linux on several of its server systems.

IT auditors can expect to encounter more and more installations of Linux when reviewing IT general controls. Many users in a Linux environment find that it has more features and flexibility than even popular Windows-based applications. When an IT auditor includes a Linux OS as part of a general controls review, he or she should ask questions similar to these:

- **Linux users.** Who can perform what functions at an OS level, and do those rights seem appropriate for the person's duties?
- **Services.** What services, server functions, protocols, modules, and other functional components are installed and/or running on the system, and are they needed and authorized? Because of the open nature of Linux, a system can be filled with many unknown software components.
- **Networks and connectivity.** What protocols and devices are allowed to reach the Linux system? What other hosts are allowed to reach this system, and do they all appear appropriate?
- **File systems.** Are only approved file systems in use, and is Linux device utilization monitored?
- **Logging/Auditing.** Are all required Linux events recorded and analyzed?
- **Security configuration.** Are appropriate user access restrictions installed and system enforced, such as to password life controls and other parameters?
- **Applications.** Are only necessary and approved applications installed and running on the Linux system?

Thousands of commands can be developed through a Linux OS, and an IT auditor should meet with enterprise IT software specialists to gain a better understanding of the implemented system. An IT auditor should look for OS system configuration standards and build procedures, jointly defined by enterprise system administrators and information security groups. Although some IT auditor technical guidance may be needed, an effective audit process would then consist of a comparison of those standards to the current configuration of the Linux system. These detailed procedures are beyond the scope of this chapter, but an IT auditor can ask a systems administrator to demonstrate compliance.

IT auditors will encounter more and more Unix and Linux installations as they review general controls in larger server systems. They should take the time and effort to become familiar with these very important OS tools.

FEATURES OF A COMPUTER OPERATING SYSTEM

An IT OS acts as an interface between its applications and the hardware. A key element of that hardware, particularly on a personal or microcomputer, is the circuitry to operate the computer's functions. This is known as the central processing unit (CPU), the circuit chip that controls the machine. The CPU's logic provides services for key processes, such as assigning memory and other resources, establishing task priorities for the process, loading program code into memory, and executing programs.

These programs then interacts with the user and/or other devices and performs its intended function. Although IT auditors typically do not need to understand the detailed functions of any OS, whether Windows XP, Linux or others, every OS has the same general functions:

- **Interrupts.** A central element of any OS, interrupts are a mechanism for the OS to interact with and react to its environment. Interrupts provide a computer's CPU with a way of automatically running specific code in response to events. Most CPUs and their OS support hardware interrupts that allow the programmer to specify code that can be run when that event takes place.

  When an interrupt is received, the computer's hardware automatically suspends whatever program is currently running, saves its status, and runs computer code previously associated with the interrupt; this is analogous to placing a place marker in a book in response to a phone call. Interrupts may come from either the computer's hardware or the running program.

  When a hardware device triggers an interrupt, the OS decides how to deal with the event and establishes protocols for doing so. This is similar to how a person usually responds to what may be a false smoke detector alarm in the home before just calling for help. The processing of hardware interrupts is a task that is usually delegated to software called device drivers, which may be part of the OS, part of another program, or both. A program may also trigger an interrupt to the OS. If a program wants to access hardware, for example, it may interrupt the OS, which causes control to process the request.
- **Memory Management.** An OS is responsible for managing all system memory which is currently in use by its programs. This ensures that a program does not interfere with memory already used by another program. Since programs time-share, each program must have independent access to memory. If a program fails, it may cause memory used by other programs to be affected or overwritten. Malicious programs, or viruses, may purposefully alter another program's memory or may affect the operation of the OS itself. It can take only one misbehaving program to crash a computer system. Various methods of OS memory protection exist, but all require the particular and unique features of each CPU.
- **Virtual memory.** Desktop and laptop computer CPUs generally do not have enough available random access memory (RAM) to run all of the programs that most users expect at once. Virtual memory is an operating system feature that enables a system to use RAM memory address space that is independent of other processes running in the same system, and to use a space that is larger than the actual amount of RAM present, temporarily relegating some contents from RAM to a disk, with little or no overhead. Virtual memory looks at RAM for areas that have not been used recently and copies them onto spaces in the hard disk to free up space in the RAM to load other applications. Exhibit 8.3 shows this virtual memory concept.
- **Multitasking.** The important OS concept of multitasking refers to the running of multiple independent programs on the same computer, making it seem as if the computer is performing multiple tasks at the same time. Since most computers generally can only do one thing at one time, OS time-sharing concepts are used.

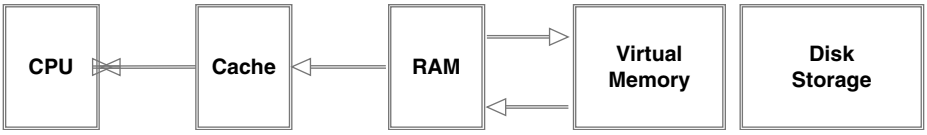

EXHIBIT 8.3 Virtual Memory Management Concepts

Each program uses a share of the computer's time to execute the program, using a piece of the OS called a scheduler, which determines how much time each program will spend executing and in which order execution control should be passed to programs. Control is passed to a process that allows programs access to the CPU and memory so that another program can use the CPU.

- **Disk access and file systems.** Access to data stored on disks is a central feature of all OSs. Computers store data on disks using files, which are structured in specific ways in order to allow for faster access, higher reliability and to make better use of the drive's available space. An OS defines the specific way in which files are stored on a disk and enables them to have names and attributes.

  Operating systems generally support a single type of disk drive and only one kind of file system. These file systems have been limited in their capacity, speed, and the kinds of file names and directory structures they can use. These limitations often reflected limitations in the OSs they were designed for, making it very difficult for an OS to support more than one file system. However, Unix and Linux support a technology known as a virtual file system (VFS). Unix supports a wide array of storage devices, regardless of their design or file systems to be accessed through a common API. This makes it unnecessary for programs to have any knowledge about the device they are accessing. A VFS enables the OS to provide programs with unlimited access to a number of devices with an infinite variety of file systems installed on them through the use of specific device drivers and file system drivers.
- **Device drivers.** A device driver is a specific type of software that allows interaction with hardware devices. It is an interface for communicating through what is called a computer bus or communications subsystem, providing commands to and/or receiving data from the device and on the other end, from the requisite interfaces to the OS and software applications. It is an OS-specific hardware-dependent computer program that enables an OS, applications software package, or computer program running under the OS to interact transparently with a hardware device. It usually provides the requisite interrupt handling necessary for any necessary asynchronous time-dependent hardware interfacing needs. An OS essentially dictates how every type of device should be controlled. The function of the device driver is to translate these OS-mandated function calls into device-specific calls. In theory, a new device that is controlled in this manner should function correctly if a suitable driver is available.
- **Networking.** An OS supports a variety of networking protocols, hardware, and applications for using them. This means that computers running dissimilar OSs can participate in a common network for sharing resources such as files, printers, and scanners using either wired or wireless connections. Networks can essentially allow a computer's OS to access the resources of a remote computer to support the same

functions as it could if those resources were connected directly to the local computer. These network connections include everything from simple communication, to using networked file systems or even sharing another computer's graphics or sound hardware.

Client-server networking involves a program on a computer somewhere that connects via a network to another computer, called a server. Servers offer various services to other network computers and users. These services usually are provided through ports or numbered access points beyond the server's network address. Each port number usually is associated with a maximum of one running program, which is responsible for handling requests to that port.

- **Security.** Computer security depends on a number of technologies working properly, but the OS provides access to a number of resources that are available to software running on the system and to external devices, such as networks. An OS is capable of distinguishing between requests that are allowed to be processed as well as others that should not be processed. An OS may distinguish between "privileged" and "nonprivileged" software transactions, and systems commonly have a form of requester *identity*, such as a username. To establish identity, there may be a process of *authentication*. Often a defined username must be entered, and each username may have an associated password as well. Other methods of authentication, such as magnetic cards or biometric data, might be used instead. Security features, discussed in Chapter 19, are a very important component of an OS.

OTHER SYSTEMS SOFTWARE TOOLS

Exhibit 8.1 shows a conceptual view of a computer system, with the CPU, RAM, and disk storage devices in the center and surrounded by the OS. The next ring outside of the OS but before user applications includes systems software tools. The processes in this ring are such things as database management systems and a wide variety of software tools, including programs that are not part of the OS but also not applications. This category of software includes utility programs for such areas as utilities for file sharing, print services, e-mail, Web sites, and what are called file transfer protocols (FTPs).

Many of these utility programs have evolved because the OS for some computer systems did not have some needed services or because outside providers developed superior solutions. As an example here that dates back to the very early days of computer systems when early IBM mainframes and their COBOL program compilers did not have a very good way to sort a program's data. Other machine vendors had built efficient data sort routines into their COBOL compilers, but the major vendor at that time, IBM, had not.

To help with this weakness, a group of New York University students developed a sort program in 1968 that eventually became SyncSort (www.syncsort.com), a well-respected utility program that became almost a standard attachment for IBM mainframe applications. Although many years have now passed, a much-enhanced SyncSort is still used frequently. A large number of utility programs—by IBM and other vendors—became common in IBM mainframe systems. The IT auditor who encounters these

utility programs should ask questions about the reasons for their use and assess whether they seem appropriate.

IT auditors encounter these and many other utility programs as help or service programs for their own laptop machines. Many are excellent, and often they can be downloaded at no charge or for a small license fee. In many cases for Windows- or Mac-based utility software, the actual programs today reside on the Internet, and the download only provides communication links. Perhaps the best example of such free utility software can be found with Google, with its search engine, communications software, and more. Google is a very reputable company; the software is free but includes some advertising messages.

A large body of utility programs, particularly for Windows or Mac systems as well for Unix and Linux servers, have been released by small or little-known vendors that may contain security risks or other control concerns. IT auditors should always raise questions and express concerns, when they encounter any such software. Exhibit 8.4 contains review guidelines for an auditor's search for unauthorized software. Procedures will vary by the type of OS, but an IT auditor should focus on authorization rules for installing any software as well as security rules for the software package.

Many variations of utility software may be installed on any system. In the next sections we focus on several basic software types as well as some of the risks of using unauthorized versions. The text presents only a small sample of the many software types available. In our era of freeware software, university labs or similar sources may offer many versions. Such products may be technically brilliant in design but may never advance in maintenance and upgrades. IT management and certainly IT auditors should be aware of potential risks associated with such utility software.

1. Review and document the software server environment, gaining an understanding of the persons or authorities responsible for authorizing and approving utility software.
2. Determine whether there is a comprehensive inventory of installed software in place, including the authorizing authority for the software and other supporting documentation.
3. Determine that the enterprise maintains a roster of approved supported software.
   a. Review documentation and controls for that supported software.
   b. Determine that appropriate cost-benefit analyses or other justifications have been developed to document the supported software.
4. Determine whether there is a process in place to regularly review software licenses and updates.
   a. On a test basis, select several installed software packages to ensure their licenses are up to date.
   b. Also on a test basis, trace several installed software packages back to their original purchase justification to determine whether supporting documentation and approval appear appropriate.
5. Identify any installed unlicensed software, and make recommendations to either remove or uninstall the software.
6. If any software has been installed as freeware, determine that it was justified and approved, that the IT organization has assessed and improved its internal controls, and that the software is still appropriate to enterprise operations.

EXHIBIT 8.4 Unauthorized Utility Software Review Guidelines

File Transfer Protocol Software

FTP is a common network utility software tool used to exchange and manipulate files over an Internet-based networks. FTP is based on client-server architecture, using separate control and data connections between the client and server applications. It is also often used as an application component to automatically transfer files for program internal functions and can be used with user-based password authentication or with anonymous user access.

This software was first developed in the very early days of dial-up IT telecommunications, and it was originally an unsecure method of transferring files because FTP had no method for data encryption. Thus, under earlier network configurations, usernames, passwords, FTP commands, and transferred files could be captured by anyone on the same network using what is called a packet sniffer detection tool.

FTP security has since been improved, and FTP processes have been built into most Web servers. There are multiple versions of this software available for transferring files. The IT auditor should attempt to assess where versions of FTP have been installed on a system, as part of a general controls review, and also should understand that this software can be used to improperly copy and capture critical data files.

Virus Protection Software

Software virus detection and other cybersecurity software tools are discussed in Chapter 20. This is a major category of utility software, no matter what computer platform or OS, because of the many threats facing all systems. There are some cybersecurity tools built into some OS versions, but an IT auditor should see effective implementation of this an important category of utility software as part of any general controls review.

Multiple vendors offer virus and IT security protection software. Because of the complexity of detecting cybersecurity problems and correcting them, a software vendor needs a strong staff to stay ahead in this ever-changing area. Some very effective freeware software products are available as well as multiple commercial packages. An enterprise must select a product that appears to suit its needs. Most versions of this software are effective, but the software must be updated almost constantly to counter new virus risks that have been identified.

We discuss these cybersecurity risks and threats in greater detail in Chapter 20. An IT audit should verify that the current version of such software covers all systems platforms, that the software is updated on a regular basis, and that there is follow-up to correct and repair any reported violations.

Automated Security Self-Evaluation Tools

A variety of tools are designed not for IT auditors but for enterprise IT management to assist in improving IT operations. One example is the Automated Security Self-Evaluation Tool (ASSET) from the U.S. National Institute of Standards and Technology (NIST). This freeware software gathers data, generates reports, and provides a centralized place for the collection of system access data, as do other self-evaluation

Network and Systems Management Software Tools*	*
Network Management Products	35
Application Management Tools	39
Software Performance Management Software	30
General Network Monitoring Software	57
Bandwidth Traffic Monitoring Tools	9
Web Usage Monitors	17
Asset Managemement/Resource Inventory Tools	21
Software License Management Tools	9
Software Compliance Tools	37
Software SQL Help Tools	33
Installation and Deployment Software Tools	
Software Distribution Products	14
Software Packaging Products	7
Drive Imaging Tools	5
Migration and Configuration Managers	15
Server Migration Aids	15
SoftwareGroup Policy Managers	22
Software Tools for Systems Administration	
Disk Degragmentation and Drive Monitoring Tools	9
Remote Troubleshooting Products	23
Network Automation and Patch Processing Tools	20
Program Scribing Tools	14
Patch Management Software Tools	19
Application Sharing Software Tools	3
Application Conflict Testing Tools	5
Software OS Security Products	
Hardware-Based Firewall Products	23
Software-Based Firewall Products	17
Security Auditing Software Tools	40
Intrusion Detection Systems	16
Intrusion Prevention Systems	24
Smart Card/Biometric Authentication Systems	13
Spam Content Filtering Tools	67
Antivirus Software Tools	21
Antispyware Tools	17
Storage and Backup Software	
Software Backup Systems	40
Storage Management Software Tools	29
Disaster Recovery Products/Services	46
Clustering and Fallover Software	13
Load-Balancing Software	17

*Total number of Windows-related software products in each category, per *Redmond* magazine (November 2009).

EXHIBIT 8.5 Utility Software Product Categories for Windows-Based Products

tools. Our objective here is not to discuss this software but to describe it as a type of freeware.

We have presented several examples of the many types of utility programs that an IT auditor may find installed on a server system. Exhibit 8.5 is an extensive list of these general categories of software. For example, we list five different categories of installation and deployment software tools as well as vendors offering products in various categories for each. At the time of this publication, 15 different vendors offered migration and configuration management products under the category of installation and deployment software tools. Vendors will come and go over time, but there are a large number of software products out there. In addition, our list here only includes Microsoft Windows-based product and does not include other software for the Mac, Unix, Linux, and others.

Our point and warning to IT auditors is that a typical IT production system may have many of these software tools installed. Often a tool was added to a system because of a special need at one time or due to the cajoling of an overzealous salesperson. When encountering these software products as part of any IT general controls review, an IT auditor should ask:

- What is the purpose and function of the utility software, and who in the IT department is responsible for the software product?
- Is the license for the software current?
- Is the software regularly used at present?
- Are there any internal control issues or concerns associated with the software product?

There are a large number of software tools installed on any server system, and some may have served a purpose at one time but may no longer be effective. Others may have been offered by vendors that are no longer in business. An IT auditor should be aware of the types and versions of OS and OS support software installed on any system as part a general controls review. He or she can provide a genuine service to management by surveying the utility programs in place as part of a general controls review and making recommendations to update or streamline these program functions.