7.6 Operating System Patches

From time to time, security flaws have been found in operating systems. As software vendors become aware of flaws that might exist, they usually write the corrections to their code, known as patches or updates. Whatever operating system you use, you must apply these patches as a matter of routine.

Windows patches are probably the most well-known, but patches can be released for any operating system. You should patch your system any time a critical patch is released. You might consider scheduling a specific time simply to update patches. Some organisations find that it is necessary to update once per quarter or even once per month is necessary.

7.6.1 Applying Patches

Applying patches means that the operating system, database management systems, development tools, Internet browsers, and so on are all checked for patches. In a Microsoft environment this should be easy because the Microsoft website has a utility that scans your system for any required patches to the browser, operating system, or office products. It is a very basic tenet of security that ensures that all patches are up-to-date.

This should be one of your first tasks when assessing a system. Regardless of the operating system or application vendor, you should be able to go on its website and find information on how to download and install the latest patches. But keep in mind that everything must be patched—the operating system, applications, drivers, network equipment (switches, routers, etc.), literally everything.

Once you have ensured that all patches are up to date, the next step is to set up a system to ensure that they are kept up to date. One simple method is to initiate a periodic patch review where, at a scheduled time, all machines are checked for patches. There are also automated solutions that will patch all systems in your organisation. It is imperative that all machines be patched, not just the servers.

7.6.2 Automated Patch Systems

Manually patching machines can be quite cumbersome, and in larger networks, simply impractical. However, there are automated solutions that will patch all systems on your network. These solutions scan your systems at pre-set times and update any required patches.

7.6.3 Windows Update

For systems running Microsoft Windows, you can set up Windows to automatically patch your system. Recent versions of Windows have it on automatically. If your system is older, simply go to https://support.microsoft.com/en-us/help/12373/windows-update-faq and follow the instructions to keep your system updated. This will give that individual machine routing updates for the Windows operating system.

It is acknowledgeable that this approach has a few shortcomings. Firstly, it only updates Windows and neglects any other applications on your machine. The second drawback is that it does not provide any way to check patches on a test machine before deploying them to the entire network. It offers two main advantages though. First and foremost it is free. Secondly, it is integrated with the Windows operating system.

Surprisingly, another commonly overlooked protection is some type of software update platform. Windows Server Update Services (WSUS), System Centre Configuration Manager (SCCM), and other third-party applications can keep the endpoints up-to-date with the latest security patches. Not only should you worry about regular Windows system patches, but there should also be a focus on outdated versions of commonly exploited software such as Java, Adobe Reader, Firefox, and others that are currently in use.

7.6.4 Unix/Linux Software Updates

Unlike Microsoft environments, Unix-based environments typically use a system of package management to install the majority of third-party applications.

Package management and update tools vary depending not only on which flavor of Unix you are running, but also differ depending on the distribution you use. For example, Debian Linux and SUSE Linux use

two different package management systems, and FreeBSD uses another.

Despite the differences, there are common themes surrounding the package management systems. Typically, each host will hold a repository of packages that are available to install on the system via local tools. The system administrator issues commands to the package management system to indicate that she wishes to install, update, or remove packages. The package management system will, depending on configuration, either download and compile, or download a binary of the desired package and its dependencies (libraries and other applications required to run the desired application), and install them on the system.

The various package management systems are so comprehensive in a modern distribution that for many environments it would be unusual to require anything further. Deploying software via package management, as opposed to downloading from elsewhere, is the preference unless there is a compelling reason to do otherwise. This greatly simplifies the issue of staying up-to-date and tracking dependencies.

The same package management system can be used to perform upgrades. As the repository of available packages is updated, new versions of already installed packages appear in the package database. These new version numbers can be compared against the installed version numbers and a list of applications due for an upgrade to a new version can determined automatically, typically via a single command line.

This ease of upgrade using package management means that unless a robust system of checking for and applying changes is in place for installed applications, the package management system should be used to provide an easy, automated method of updating all packages on UNIX application servers.

Not only does this remove the need to manually track each application installed on the application servers, along with all their associated dependencies, but it (typically) means that it has already been tested and confirmed to work on that distribution. Of course, individual quirks between systems mean that you cannot be sure that everything will always work smoothly, and so the testing process should remain.

However, the testing process may be entered with a good degree of confidence.

7.6.4.1 Core Operating System Updates

Many, but not all, UNIX systems have a delineation between the operating system and applications that are installed on it. As such, the method of keeping the operating system itself up-to-date will often differ from that of the applications. The method of upgrading will vary from operating system to operating system, but the upgrade methods fall into two broad buckets:

Binary update

Commercial operating systems particularly favour the method of applying a binary update; that is, distributing precompiled binary executables and libraries that are copied to disk, replacing the previous versions. Binary updates cannot make use of custom compiler options and make assumptions about dependencies, but they require less work in general and are fast to install.

Update from source

Many open source operating systems favour updates from source, meaning that they are compiled locally from a copy of the source code and previous versions on disk are replaced by these binaries. Updating from source takes more time and is more complex, however the operating system can include custom compiler optimizations and patches.

There are many debates over which system is better, and each has its pros and cons. For the purposes of this book, however, we will assume that you are sticking with the default of your operating system as the majority of arguments centre on topics unrelated to security.

Updates to the operating system are typically less frequent than updates to third-party software. Additionally, they are more disruptive, as they typically require a reboot because they often involve an update to the kernel or other subsystems that only load at startup, unlike application updates, which can be instantiated via the restart of the appropriate daemon. Core operating updates are advisable even though they are often found as vulnerabilities within both operating systems and applications.

As with any other patch of this nature, it is advisable to have a rollback plan in place for any large update such as one for an operating system. In the case of virtualized infrastructure, this can be achieved simply by taking a snapshot of the file system prior to upgrade; thus a failed upgrade can be simply rolled back by reverting to the last snapshot. In physical infrastructure this can be more problematic, but most operating systems have mechanisms to cope with this issue, typically by storing a copy of the old binaries and replacing them if required.

Nevertheless, patches to the operating system are often required in order to close security gaps, so you should have a process defined to cope with this. As with applications, the effort to upgrade the operating system is lower than a more up-to-date system already is. We recommend remaining as current as possible, leaving only small increments to update at any time.