How to Make Microsoft Security Patch Testing More Efficient

Abstract

Despite the perils of deploying a Microsoft® security patch to an enterprise’s operating environment without first assessing its impact, many organizations today still fail to conduct sufficient predeployment patch testing. Although there are many challenges to instituting an efficient patch testing process, solutions are available that can reduce your testing time and make the workload much easier to handle. This document explores the best practices for making patch testing not only feasible, but more reliable, efficient, and effective – and the solutions available for facilitating the implementation of these best practices.
# Table of Contents

**Introduction** 3

**Acquiring Microsoft Security Patches** 3
  - The Second Tuesday of the Month 3

**The Importance of Patch Testing** 5
  - The Perils of Deploying Untested Patches 5
  - Why Organizations Don’t Test Security Patches 5

**Making Security Patch Testing More Efficient** 7
  - To Test or Not to Test – That Is No Longer the Question 7
  - Step One: Understanding Your Application Landscape 7
  - Step Two: Narrowing Your Patch Testing Focus 7
  - Step Three: Investigating Detected Impacts 8
  - Step Four: Handling Incompatible Patches 8

**Summary** 9
Introduction

In May 2004, the Sasser Internet worm crippled computer networks at business, transportation, and government agencies in North America, Europe, and Asia, costing organizations an estimated $15 billion in lost business and repair expenses. The worm spread rapidly by infiltrating a security breach in Web-connected computers that use Microsoft’s Windows operating system. Despite the chaos and expense it caused, perhaps the worm’s most insidious quality from a systems administrator’s standpoint was how quickly it exploited a Microsoft-detected OS vulnerability: Sasser first appeared a mere 18 days after Microsoft announced the patch for the security hole Sasser would use.

In the past, it often took hackers several months to exploit a new vulnerability, but the amazing swiftness of Sasser delivered what for many IT departments was a wake-up call they can no longer ignore: patch faster. For today’s businesses, instituting an efficient patch management strategy is no longer a luxury. IT security failures caused by viruses and worms such as Sasser and its predecessors, Blaster, Welchia, Slammer, and Nimda, cost businesses billions of dollars each year, and the window of time organizations have from the moment Microsoft releases a patch for a detected vulnerability to the first launch into the wild of an exploit by a hacker is becoming increasingly small. Expedient security patch acquisition and deployment has never been more critical.

Unfortunately, simply deploying a newly acquired security patch to your operating environment without first testing its impact on your systems can have potentially catastrophic results. A single incompatible patch can crash mission-critical applications and destabilize workstations across your entire organization, causing even more damage than the virus it was deployed to stop. For many organizations, however, effective patch testing can be a time-consuming, and often neglected, process. This document explores how to simplify the predeployment testing of Microsoft security patches by narrowing your testing scope and discusses the options for handling those patches that negatively impact your organization’s applications.

Acquiring Microsoft Security Patches

The Second Tuesday of the Month

Systems administrators managing workstations and servers that run Microsoft operating systems know the significance of the second Tuesday of every month, the day Microsoft releases its list of new OS security patches. On this day, systems administrators responsible for their organization’s patch management effort must decide which of the new security patches they need to acquire and deploy to relevant areas of their operating environment.

Several solutions exist to assist you in managing the acquisition of Microsoft patches, including Windows
Update Services, Shavlik™ HFNetChkPro, PatchLink™, and BigFix® Patch Manager. Organizations not using one of these solutions can use FLEXnet® AdminStudio® from Macrovision® to simplify the acquisition of Microsoft patches. FLEXnet AdminStudio creates a real-time list of every available Microsoft patch – not just security patches – and enables you to sort them by OS version and service pack. You can also see at a glance which patch data you have previously downloaded and the patch data you have yet to acquire.

FLEXnet AdminStudio makes it easy to obtain and manage Microsoft patches.

It is important to note that FLEXnet AdminStudio works as a complement to the patch management solutions mentioned above, not as a replacement. It offers systems administrators exclusive features found in no other product that work in conjunction with other patch management solutions to help complete an organization's overall patch management effort. More information on FLEXnet AdminStudio will be provided later on in this white paper.
The Importance of Patch Testing

The Perils of Deploying Untested Patches

"Testing the impact of a patch on your operating environment before deployment is a vital step in any successful patch management strategy. Skipping this step can result in service disruptions and can require significant overtime to repair incorrectly modified machines."

Fred Broussard
IDC

"It is recommended that all security patches be tested to a test bed of servers and workstations matching your corporate business environment to ensure compatibility issues do not exist."

Microsoft
TechNet, July 2003

Once new Windows security patches are obtained, most organizations will deploy the patches to their environment as quickly as possible without conducting any significant predeployment testing. As stated earlier, this is a dangerous move. Because every Windows OS patch has the potential to modify the runtime file dependencies of one or more of your enterprise’s applications, deploying a patch without knowing how it will affect every application is very risky. A single incompatible patch can cause multiple applications failures and malfunctions, impairing an organization's ability to do business and forcing IT to waste time and resources correcting the problem. The results can range from a minor inconvenience, if one or two lightly used applications are affected on a limited number of workstations, to a major disaster, if a single mission-critical application is brought down for even a day.

Why Organizations Don’t Test Security Patches

There are several reasons why organizations forgo predeployment patch testing altogether and choose instead to rely on the so-called “deploy-and-pray” patch management model:

- **Unknown Operating Environment.** Although this may sound strange to IT personnel managing a small number of desktops and servers, many systems administrators simply do not know what applications are installed in their operating environment. Systems administrators in large organizations can be responsible for upwards of 5,000 desktops, each running their own set of third-party, customized, and internally developed applications. If the organization has failed to institute and enforce a set of desktop standards, it is difficult for a systems administrator to know exactly what is on each workstation, thereby making patch testing far more challenging.

- **Limited Testing Resources.** Even organizations with tightly managed desktops may take the
deploy-and-pray tactic with their patches because they lack the IT resources to conduct proper testing. If an organization’s ratio of applications to administrators is over 200 to 1, knowing exactly which applications and operating system version are on every workstation and server does little good. Thoroughly testing patches against every application in the operating environment, especially during months with a heavy volume of new patches, remains an unfeasible task. Doing so manually would grind all other IT activity to a halt.

- **Time Constraints.** Time is not a system administrator’s friend, especially in regards to patch testing. Once Microsoft posts a new patch, the clock starts ticking. Considering the alarming speed with which today’s hackers can exploit new reported vulnerabilities (Blaster took 30 days to appear, while Sasser took only 18), IT’s timetable to deploy new patches should be 30 days or less from the moment Microsoft posts the patch. Any delays or bottlenecks could potentially cost a company millions of dollars in downtime. While testing multiple patches against every application in an organization is strenuous enough, the 30-day deadline makes it that much harder, especially when a whole new batch of security patches requiring testing comes every month.

- **Blind Patch Deployment.** Some administrators believe that deploying a Microsoft security patch as soon as it is acquired is worth the risk. They know that deployment delays leave their organization vulnerable to attack, and they believe the risk of not deploying immediately outweighs the potential damage an untested patch may cause to their common operating environment. Plus, these administrators may assume that since a patch is from Microsoft, it can be trusted and isn’t in need of testing.

Other organizations compromise on speed or testing quality using one of the following approaches:

- **Patch Deployment Anxiety.** An alternative to the deploy-and-pray tactic employed by IT departments is to not deploy the patches at all but to wait for Microsoft to release their next OS service pack. Service packs are usually more stable than patches, and the risk of deploying them, even to a poorly managed operating environment, are typically lower. However, there is no set time table for the release of OS service packs, making the “wait-and-pray” tactic extremely risky, as it leaves an organization vulnerable to multiple catastrophic attacks. For a systems administrator, having to explain to the CEO the logic behind not deploying a patch that could have prevented millions of dollars in damage and lost productivity caused by a worm infestation could be a career-ending experience.

- **Minimal Patch Testing.** Some organizations that conduct predeployment tests on their security patches do so only against a baseline image of their environment, usually consisting of the OS and some core set of applications (often Microsoft Office and Adobe Acrobat Reader). These tests, while easier for an IT department to conduct, fail to take into account the wide variety of
third-party and custom mission-critical enterprise applications found throughout an organization’s operating environment. Unfortunately, the untested custom applications are those that organizations can least afford to have go down for any length of time.

Making Security Patch Testing More Efficient

To Test or Not to Test – That Is No Longer the Question

Fortunately for systems administrators, debating whether or not to test newly acquired Microsoft security patches before deployment is no longer necessary. Organizations no longer have to rely on the deploy-and-pray model, wait for the service pack, or conduct cursory predeployment tests on incomplete images. Best practice steps exist that make patch testing not only feasible, but reliable, efficient, and effective – and solutions are available for facilitating the implementation of these best practices. The best practice steps and their accompanying solutions are discussed below.

Step One: Understanding Your Application Landscape

Organizations lacking an inventory of the applications on their desktops and servers can never truly institute an effective patch testing strategy, so the first step is to gain a complete understanding of the contents of every workstation and server across the entire enterprise. System management solutions such as SMS 2003, ZENworks™, and LANDesk™ all have tools for creating an inventory of all the software assets in your operating environment. Inventories can be created manually as well.

Step Two: Narrowing Your Patch Testing Focus

Once you know which applications are present on the desktops and servers targeted by new Microsoft security patches, you still have to test the patches against each of the applications (which could number in the hundreds), and you have little time and resources to do it. So, after capturing a complete inventory of your operating environment’s assets, the next step is to find ways to exclude as many of those assets as possible from your actual patch testing.

Many systems administrators use FLEXnet AdminStudio solution to better focus their patch testing efforts. FLEXnet AdminStudio has the exclusive ability to detect – without actually deploying the patch – if a patch will impact the runtime dependencies of any application in your operating environment. If a patch’s deployment will in fact replace one or more Windows DLLs used by your enterprise’s applications during runtime, FLEXnet AdminStudio will list which applications are affected. So, instead of trying to test every application in your enterprise against every security patch you acquire, you now only have to test the three or four applications that will actually be impacted. What was once a difficult task now becomes something that even organizations with limited IT resources can accomplish in a reasonable amount of time.
FLEXnet AdminStudio provides Web-based reports that list every impact of a patch’s deployment on an organization’s applications.

Step Three: Investigating Detected Impacts

Once FLEXnet AdminStudio has identified what, if any, applications will be impacted by a new patch’s deployment, the next step is to do the actual patch testing. To properly test a patch, you should set up a clean machine that mimics each environment where the impacted applications are running (including service packs, operating system, applications, and antivirus software) and install the patch to that clean machine. Once the patch is installed, run the impacted applications to see if the patch has negatively affected their performance and behavior. If all the applications function properly, the patch is safe to deploy. If any of the applications malfunction or crash, the patch is incompatible.

Step Four: Handling Incompatible Patches

If you discover that a security patch will negatively impact the performance of one or more of your enterprise’s applications, you have another option before deciding whether or not to deploy the patch to those desktops or servers running the impacted software: isolate the impacted applications.

FLEXnet AdminStudio facilitates the isolation of an application so that it always loads the versions of
components – such as DLLs – with which it was originally developed and tested. This is accomplished by providing DLLs and other shared components for the impacted application and placing information traditionally stored in the registry into manifests that specify the location of these components. By successfully isolating the application, the incompatible patch would be cleared to alter the OS files without affecting the isolated software.

It is important to note, however, that not every detected impact can be resolved through isolation. When this occurs, systems administrators need to carefully weigh the business value of the impacted application versus the significance of the vulnerability corrected by the security patch. If the application is too valuable to the enterprise, delaying the patch’s deployment to those desktops using the application may be worth the security risk of waiting for the next service pack. In these cases the IT department would want to implement additional layers of security to limit the vulnerability and exploitability of the critical, unpatched environment. Some options include ensuring that:

- The unpatched machines are behind multiple firewalls that employ a “nothing comes in, nothing goes out” configuration
- Unpatched laptops are separated from the rest of the network by a properly configured firewall
- File types are filtered on mail servers and .exe, .vbs, and .pif files are automatically deleted

As a final note, it is critical that you avoid the temptation to alter the Microsoft patch directly. Deploying a modified patch could have catastrophic results on your enterprise’s applications and still fail to plug the vulnerability it was created to eliminate. It could also negatively impact the ability of future Microsoft security patches to eliminate vulnerabilities, since those new patches would be anticipating a specific environmental configuration that your environment, due to the modified patch, would no longer match.

Summary

For many organizations, taking time to conduct proper predeployment testing of Microsoft security patches is viewed as an overwhelming task. Although there are many challenges to instituting an effective patch testing process, solutions are available that can significantly reduce the time it takes to properly test each patch before deployment. FLEXnet AdminStudio makes testing all Microsoft patches – not just security patches – a far more manageable and efficient task. To learn more about how FLEXnet AdminStudio simplifies the management of your applications and patches, or to download an evaluation version, visit http://www.macrovision.com/products/flexnet_adminstudio/adminstudio.