BAROUDI BLOOR

THE PATCH PROBLEM

It's Costing your Business Real Dollars

175 Pleasant Street Arlington, MA 02476 617-747-4045
www.baroudi.com

Introduction

Bank of America and Washington Mutual ATMs out of service, Continental Airlines flights cancelled, the City of Seattle's 911 service not working all because an administrator failed to apply a patch that had been available for six months.[1] It sounds like the stuff of a big-budget disaster film, but it's not. It's the actual damage done in 2003 by the SQLSlammer worm, also known as Sapphire. This worm proves that the "Patch Problem" isn't just a nuisance; it's a bona fide disaster waiting to steal real dollars from a company's profit line.

The need for reliability is taken for granted in endeavors outside of computer systems and networks. If you're not feeling well on a prescribed medication, you call the doctor and ask to have the prescription changed. If your car is put on a recall list for brake maintenance, there's no hesitation, you make an appointment with the garage and go down to have the problem repaired. But if a patch is released for a mission critical server, many administrators will not apply it. This cost of not applying patches can mean real dollar loss for organizations, so why does it happen?

It is partly due to our acceptance that computer systems do not always run reliably. The all too common exclamation: "Oh, the system is down" or "The system is slow" is uttered on an almost daily basis at most companies. But let's get back to that car on the recall list. Car owners not only expect their cars to be reliable, they demand it. If the brakes aren't working, the car is considered unsafe to drive. And if a recall is issued, owners know that their lives could be at risk if they do not have the problem corrected. A production server that isn't running quite right because a patch or fix hasn't been applied, however, is considered business as usual.

The complexity of the systems themselves contributes to problem. A car in recall is the same car the manufacturer released from the assembly line; the parts are known. But an operating system running in production has been tweaked, loaded with software, and is running on any number of different pieces of hardware. It's an unknown. And applying a patch could result in anything from a simple error message to something that brings the system to a complete halt or 'blue screen'.

[1] http://news.com.com/2009-1001-983540.html?tag=fd_lede2_hed

The problems associated with patches pose potential cost issues for enterprises. The question becomes, apply the patch and risk a system failure or don't apply the patch and risk attack or other inadvertent failure? Recent research indicates that 7 out of 10 systems administrators did not fix a major software bug when it was first announced.[2] And a worrisome 3 out of those 10 still failed to patch the problem even after the vulnerability had been exploited to spread a damaging worm. Another sometimes unseen cost is liability and accountability. If you are in an accident due to faulty brakes, the car manufacturer is liable and remuneration can be sought. If a server is attacked because a patch hasn't been applied, who is at fault?

What's a company to do? Apply intelligence. An enterprise can better understand the level of need for the patch by answering questions such as: does it fix a critical vulnerability and will the patch improve performance. This information can give a company a sense of how necessary the patch is. Another piece of intelligence related to patch application is the length of time since its release. If a patch is buggy or likely to cause system failure, it will most likely be discovered, and in some cases recalled, in the first 30 days of release. Therefore, waiting to apply the patch can decrease the likelihood of the patch failing. Unfortunately, there aren't many sources of information available to administrators to help them apply intelligence to their patch application strategy. This paper explores the hard dollar risk associated with patching/not patching and discusses ways in which companies can reduce that risk and keep their systems running as reliably as we expect our cars to.

The Business Case for Reliable Systems

Are reliable systems really a mission critical need? Why not just ignore the patches if the system is running? In other words, if it ain't broke, why fix it? The response, of course, is that even though a system may appear to be running well, if there is a required patch for anything from improved performance to correcting an existing security vulnerability, the system is not running optimally.

[2] http://www.newscientist.com/news/news.jsp?id=ns99993090

Before the Internet, the risk of not patching a system was limited by the very fact that computers weren't connected to each other and so could not come into contact with potentially harmful pieces of code. The risks increased dramatically, however, as enterprises built LANs and got online. Today, due to the frequent interactions and exchanges between computers within an organization and even around the world, a virus or known vulnerability can be exploited and spread in a matter of minutes.

If an enterprise cannot rely on its systems to run as expected, there is a problem that can directly affect the bottom line. Just as you need to know the brakes on your car will work when you depress the pedal, companies must be able to rely on their systems. And companies that need to engender trust are even more vulnerable. An attack on a trusted news outlet such as USA Today or on a bank damages the ability of their customers' to trust in the reliability of their services. Systems must be reliable or the business cannot succeed, and one of the most important factors in keeping systems reliable is keeping them up to date and well patched.

The Patch Problem

So if patches are such a great thing, why doesn't everyone apply them as soon as they're available? Because there are a number of very real and valid reasons for not applying them. The core requirement of a business is to keep the business running, and applying a patch can seem like an extraneous task, often viewed as a cost sink rather than an investment in the reliability and profitability of the business.

Because of this, there often aren't enough hands/staff to go around and apply the patches. Overloaded IT departments are scrambling to put out fires on a daily and even hourly basis. Time for basic management, such as patching, is scarce and the priority drops in the administrator's schedule. Since not all patches are critical, senior systems administrators need to research and analyze the importance of the patch and any associated exploitation, assess the criticality of affected servers, prioritize the need for patch application, and even contact colleagues at other companies to compare notes. Doing all of this work adds a tremendous overhead to the administrator's time burden.

Adding insult to this injury is the fact that there are so many patches being released. Microsoft alone has been known to release up to 5 per week.[3] And the patch release schedules are on the rise. Not only is software getting more complex and therefore harder to QA properly before release, but also vendors are under extreme market pressure to get their products out to the market as quickly as possible, which squeezes the QA process even more. And, of course, the attackers themselves are getting more resourceful and better able to find and exploit vulnerabilities. The result is that buggy or vulnerable code and patches are shipped to consumers. As a consequence, this under tested code frequently requires further patching after installation by the end user.

The method of distribution of the patches can also be a significant deterrent to timely installation. Some vendors ship their patches to their customers automatically, but the end-user is flooded with patches and doesn't know which ones need to be applied immediately, if at all. Some patches are available on the vendor website but are not shipped automatically to the end-user. The consumers, however, are so busy that they don't have the time to go to the vendor site to check for patches on a regular basis. Finally, some patches may not be available automatically, either via automated downloads from the vendor or on their web site. No matter how patches are distributed, there is a major potential gap in the reliability of systems.

Another reason administrators may delay patch application is the very real risk of failed patches. If a patch causes a shut down or blue screen of the system, downtime results. If the patch is applied to a critical system in production, the downtime can be very expensive to the organization. Imagine if your email administrator applied a patch to the email server that brought the system down for a few hours and no one at the company could send or receive mail. Or, if the patch was applied to a corporate portal site and access to data like sales leads and product information was cut off from business partners, employees, and customers. To reduce the risk of applying buggy patches, administrators can devote time to testing and review, but that time is then lost and man-hours are increased.

[3] http://searchwindowsmanageability.techtarget.com/originalContent/0,289142,sid33_gci860091,00.html

In some cases the patch itself is fine but it affects proprietary applications negatively. This can result in a need for development hours to re-code the application so that it works with the patch. Development time is also required if the patch causes a change to existing code that must then be updated or fixed to keep the systems running

One way to mitigate the downtime risk is to stage the patches on non-production servers to test their viability. This approach has a few drawbacks, however. It is a real cost of employee time to run the staging and the company must also pay to have staging servers for patch testing. Even more concerning is that few systems are exactly alike, and even with the investment in patch staging, administrators will still find that some patches, when applied to production systems, cause failure.

Cost to the Business

Yet there is a very real risk of costly failure if the patches are not applied. The costs can be devastating. Code Red, one of the most well known worms, was able to propagate only because IIS admins failed to apply the known patch/fix for the vulnerability[4]. How expensive was it for companies that failed to apply the patch/fix? Very. Estimates of the hard dollar damage done by Code Red are in excess of 2.6 billion USD with a phenomenal 359,000 computers infected in less than 14 hours of the worm's release.[5]

While every business has different metrics for patching costs, it is possible to make some generalized predictions in terms of the hard dollar effect on the bottom line that can be used to understand the benefit of patch management.

Breaking down the components to simple numbers, an equation emerges:

S + D + L + R = Cost of Failure to Patch

Where:
S= System Admin Time Costs
D= Development Costs
L= Lost Revenue due to Down Time
R= Reputation and Good Will Loss

[4] http://rr.sans.org/malicious/code_red.php

[5] http://www.caida.org/outreach/papers/2002/codered/

Using numbers from an October 2000 survey[6] to quantify the costs per hour we get:

S= 205USD /hr
D= 205USD /hr
L= 1million USD /hr
R= Varies

If failure to apply a patch costs 4 hours in System Admin Time to clean up the effects and patch the system, 2 hours in Developer Time to re-code any applications that have been effected by the patch or damage done by failure to patch and 30 minutes of downtime the cost of not patching is a whopping:

$820 + $410 + $500,000 = $**501,230**

This does not even take into account the very difficult, but real, cost caused by loss of reputation or good will. Taking that number into account could add upwards of hundreds of thousands of dollars in damage depending on the industry and the type of system breach caused by the failure to patch.

The numbers get even more concerning when actual down time data is used. One administrator[7] at a mid-sized company interviewed for this paper reported that his company suffered 1.5 days of downtime due to an infection of Nimda that wiped out most of the company's critical servers. Total estimated hours lost was 600 man days. Or:

$205*8 /hours per day*600 man-days = $**948,000**

Close to a million dollars in actual loss for a single outbreak of Nimbda.

Other Concerns

Patches are often cumulative and failing to install a seemingly unimportant patch one month may prevent a company from being able to apply a critical security related patch the next. And the patch latency could mean a company needs to lose business while critical systems are offline and being brought up to a patch level that includes both the 'unimportant' previous patches as well as the immediate need critical ones.

[6] http://img.cmpnet.com/nc/1205/graphics/1205f13.gif
[7] Les Ward

If you think that a strong firewall is going to protect your systems, think again. Many attacks are launched from outside the corporate firewall and still succeed in gaining access to internal servers to deface/attack them as was the case with the July 2002 attack on USA Today's web servers.[8] The reason that firewalls aren't the final answer is that they must allow through traffic that appears to be valid, and attackers exploit vulnerabilities with seemingly innocuous traffic.

Why Intelligence Solves the Problem

What can we do? We can fight fire with fire. The human element is the weak link - too many systems too many patches. The solution is to use centralized intelligence and a management solution to streamline the process. Automated intelligence means leveraging resources that don't require time from in-house corporate staff. For example, an outside service that automates the testing process and rates patch availability offloads the task from the corporation and presents a prioritized plan for patch application.

Some vendors, such as Symantec, Apple, and Microsoft, have already attempted to streamline the update/patch process by offering automatic updates to systems and applications. While this trend is cause for cautious optimism and has been put to good use at the home and end-user level, it is not a perfect solution for production enterprise servers because if the patch is not well-tested it may cause system failure. It is for this reason that most companies do not allow automatic patch/updating on their critical systems.

Most patches, if they are going to cause catastrophic failure, will be recalled within a number of days. While administrators may decrease their risk by waiting to apply a patch, during that window, the systems are exposed to the very vulnerability that the patch was released to protect against. And if the vulnerability has been covered in the media, waiting a few days may not be a possibility any longer. Once a vulnerability has been distributed to the public at large, time is of the essence for patch application. Many administrators even wait for a vulnerability to be announced on CNN, for example, before considering applying a patch. As with the 'wait a few days approach' this one also leaves the enterprise vulnerable to an attack window. In other words, if the worm is already on CNN, chances are it's already working through your network too.

[8] http://www.usatoday.com/news/site-vandalism.htm

Intelligence is the answer. What administrators need is a way to prioritize the patches available for their systems, assess the risks/vulnerabilities the patches were released to correct, and track the time from release to know when the patch should be applied.

To attempt to aggregate and correlate this kind of data at each enterprise would take time, staff, and money. However, using a service that completed this work in an automated and intelligent manner on behalf of the enterprise and then generated a priority report listing the most critical patches and which systems they should be applied to would be a cost savings. The service would free the administrator's time from wading through patches, while decreasing the risk of harmful patch application.

Summary

Enterprises rely on their networked systems for all of their business needs. Banks rely on networks for their ATMs, cities for their 911 service, and companies to complete their daily business. If these systems are not reliable and available, the company will suffer hard dollar loss. Although applying patches to keep systems reliable is the right thing to do, many companies cannot due to the overwhelming number of patches available, the lack of resources, and the potential risk from applying faulty patches.

Intelligent patch management is an absolute requirement for corporations. By applying intelligence to the patching process, companies are empowered to know about the critical, available patches and apply them at the right time. The end result is that the patch process is intelligently optimized, administrative time is saved and profitability can be increased. It's a win/win proposition and a must for the industry and enterprises going forward. You wouldn't wait to get the recalled brakes in your car fixed and you shouldn't wait to patch the production servers that run your business either. If you do, you may well get "slammed" in more ways than one.

BAROUDI BLOOR

175 Pleasant Street Arlington, MA 02476 617-747-4045 www.baroudi.com