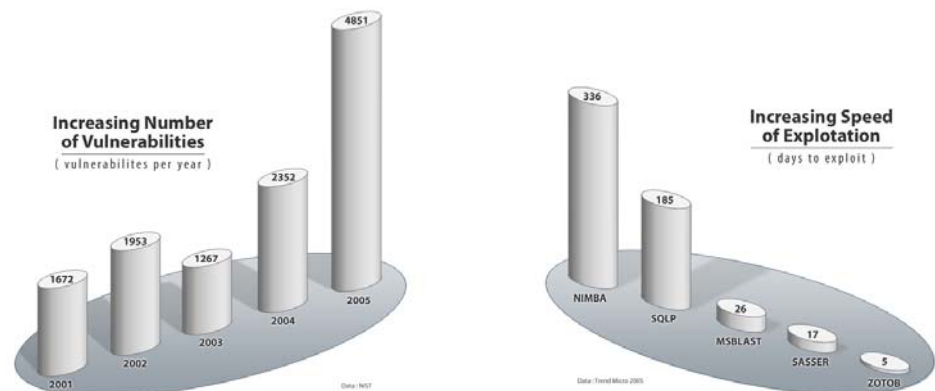




## **The Return on Investment of Automated Patch Management**

## Introduction

It's a simple truth: applying patches is the only definitive way to keep vulnerable systems from being exploited. Accordingly, the vast majority of organizations acknowledge the need to have a formal patch management strategy and solution. Furthermore they clearly recognize that the demands in this area are escalating due to the proliferation of new vulnerabilities and the rapid emergence of associated threats. Seemingly irreversible conditions require that organizations not only deploy more patches than ever before, but also that they do so with a much greater degree of urgency.



**Figure 1: A "Perfect Storm" for Information Security**

Given this situation, it intuitively makes sense to implement an automated patch management solution. However, IT and security personnel inevitably need to provide more than just their intuition to justify such an investment. This paper is intended to address this necessity by enumerating the cost savings and other associated benefits of automated patch management. Ultimately it will be demonstrated that, relative to a manual approach, an automated solution can reduce the annual cost of patching from \$222 to \$40 per computer – resulting in an expected savings of over \$180,000 per year for an organization with 1000 computers.

## Cost/Savings and Benefits Analysis

There are many factors and dependencies associated with an analysis of the benefits of automated patch management – not all of which are straightforward. The assumptions, choices, and rationale provided in the following sections are based on the experience of the authors, the expertise of the developers and engineers at PatchLink, and the continuous feedback collected from PatchLink's extensive customer base.

### Overview of Benefits

The benefits of automated patch management can be assigned to two general categories: quantitative and qualitative. The primary distinction between

these is whether reasonably defensible estimates can be calculated for the given benefit.

The most significant quantifiable benefit is the reduction in administrator effort that results from automating many portions of an otherwise manually intensive exercise. Understanding this further is facilitated by [Figure 2](#), which provides a summary of the individual tasks that comprise the major steps of a typical patch management process. To be clear, the benefit here is one of achieving greater efficiency of operations.

It could also be argued that administrator and end-user productivity gains due to incurring fewer successful attacks deserve to be classified as quantifiable benefits. However, it is probably more appropriate to classify these as red herrings. The problem in this case is that the potential gains hinge on the anticipation of remediating a vulnerability much sooner than would otherwise be possible (which is fundamentally different than doing it more efficiently). But there are several challenges with this notion. First, the presence of intermediate steps in the process which are necessarily manual diminishes the potential improvement in the overall 'elapsed time' before a patch is applied and, more importantly, complicates its quantification. The second challenge is assigning a value to whatever degree of improvement is actually attained. By how many will the number of successful attacks actually be reduced? One can only guess.

Finally, there is the point that taking advantage of any gain in this area requires the patch management process to be executed more frequently. In the extreme, it would need to be conducted every time a patch became available – as opposed to the widely favored approach of executing it at regularly scheduled intervals (e.g., monthly). Overall, it is expected that the cost of these extra cycles (i.e., rollouts) would offset the productivity savings attributable to experiencing a few less successful attacks. In any event, this potential benefit is simply too difficult to defend concretely and, therefore, is relegated to the qualitative category.

It is important to realize, however, that just because it is not easily quantified does not mean that the ability to remediate vulnerabilities sooner, at least in some cases, is not a valid benefit. In reality, it can and does save organizations from successful attacks. It's just that the actual number of such occurrences is irregular and highly unpredictable. Instead, the real value in this case is a general level of risk reduction that yields a range of qualitative benefits, such as the reduced likelihood of:

- Loss of data;
- Loss of revenue;
- Loss of credibility with customers and partners; and,
- Legal action/liability.

Furthermore, the potential magnitude of these benefits is so great that productivity gains due to fewer user disruptions and reduced recovery efforts, whatever they may be, become relatively meaningless. Indeed, it is well documented that even a single successful attack could lead to "intangible" losses of millions of dollars, particularly if the incident receives any degree of public exposure and attention.

### **Description of Scenario**

As intuitively helpful as large-magnitude qualitative benefits can be, they simply do not have the same motivational impact as cold-hard data, especially if it's in the form of dollars. With this in mind, a cost model comparing manual and automated approaches of executing the patch management process is provided in Table 1. Although this model is essentially generic, and therefore adaptable to the situation at virtually any organization, the specific scenario for which the calculations were made in this case is defined by the following high-level characteristics.

- There are 1000 end-user computing stations split among two sufficiently different builds (i.e., combinations of hardware, operating system, and applications) such that certain tasks must be performed independently for each group.
- There is a moderate level of heterogeneity, with operating systems and applications from multiple vendors. This leads to a total number of applicable patches that corresponds to twice the annual average of patches encountered by the typical Microsoft-only shop (i.e., 2\*160). However, risk analysis and shrewd planning result in the need to only deploy three quarters of this number (i.e., 240).
- The organization prefers to aggregate its patches and deploy them at regularly scheduled intervals (i.e., monthly), but will conduct additional, off-cycle rollouts to account for critical situations (i.e., 2 per year).

It should also be emphasized that estimates, where needed to supplement real-world data, were made in a conservative manner (i.e., in favor of the manual approach). As a result, the actual cost advantage that any given organization derives from automated patch management is likely to be somewhat greater than what the model predicts.

### **Examination of Findings**

Speaking of cost advantages, the outcome for the given scenario is that, due to a per-computer reduction in patch management costs from \$222 to \$40 per year, an automated patch management solution is projected to yield an annual savings of approximately \$182,000. In other words, without even accounting for any of the associated qualitative benefits, automated patch management will provide an ROI of approximately 450%, essentially paying for itself in less than three months.

Review of [Table 1](#) reveals that the largest contributions to these projected cost savings come from gains in the deployment step of the patch management process. These gains can be attributed in large part to the ability of client-side agents to minimize distribution/installation errors and to significantly facilitate any required troubleshooting.

It is also important to recognize that while deployment related tasks are responsible for the greatest degree of savings, they are not the only ones that have an impact. In fact, as can be seen in the table, modest yet still significant gains are made in each of the other steps of the patch management process as well. Particularly telling is that even these smaller gains alone are sufficient to yield an annual reduction of 940 hours of labor, resulting in savings (\$47,000) that is more than twice the cost of the patch management

solution (\$20,300). Again, a significant portion of the benefit can be attributed to the client agents. They automate both the pre-deployment task of establishing patch applicability as well as the post-deployment task of periodically validating that each patch remains properly installed. Extending beyond the patch management process, they can also facilitate inventory management objectives by identifying the software and hardware components residing on all managed systems.

### **Mileage Will Vary**

As noted earlier, the cost analysis model and resulting savings projections of Table 1 are based on a wealth of experiential data. Nonetheless, it is appropriate to acknowledge that a number of factors can impact the real-world outcome for any given organization. Some of the more significant ones include:

- Size of organization;
- Degree of centralization/de-centralization;
- Level of administrator expertise;
- Diversity of operating systems;
- Diversity of application portfolio;
- Complexity of system configurations; and,
- Enterprise policies and procedures

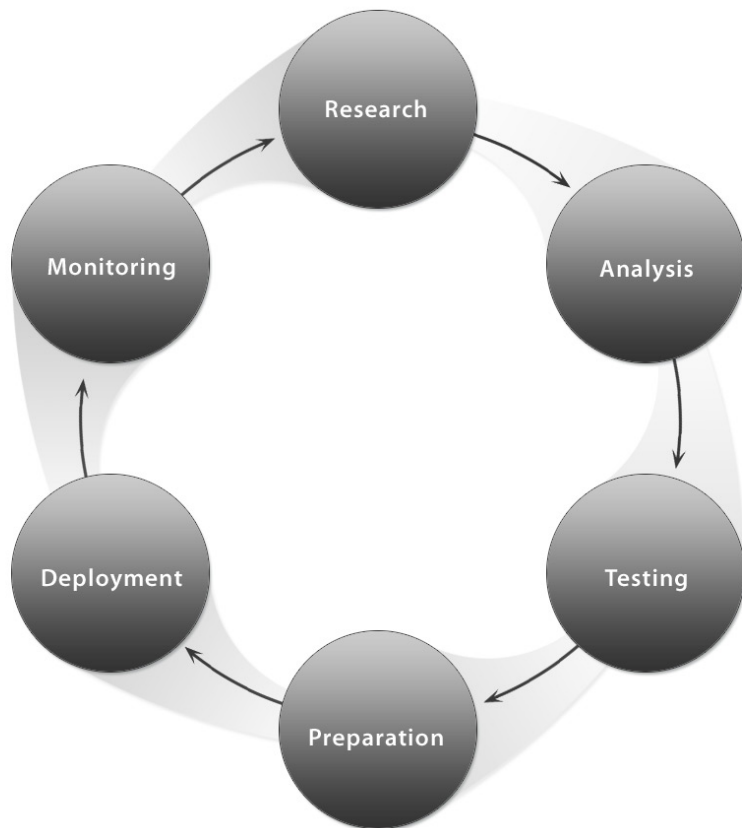
In addition, the patch management product that is selected can be another potentially significant factor. By no means are they all created equal. For example, unlike PatchLink Update, not all of them will have flexible system inventory capabilities, a streamlined patch deployment wizard, and assessment and validation services that are based on patented Patch Fingerprinting Technology. Nor will they all exhibit the advantages attributed to an agent-based architecture. For assistance selecting a best-of-breed automated patch management solution, readers are referred to the separately published whitepaper "The Top 10 Requirements for Enterprise Patch and Vulnerability Management"<sup>1</sup>

## **Summary**

In this day and age of vulnerability proliferation and fast-following threats, automated patch management is an intuitively appealing solution. The qualitative benefits alone can often be quite compelling, with better (i.e., more accurate and potentially quicker) patching leading to an overall reduction in risk as a result of incurring fewer successful attacks. In addition, for organizations seeking more concrete evidence, it can fortunately be found in the form of quantifiable cost savings. Specifically, it is expected that an enterprise patch management solution featuring a high degree of automation will reduce the annual cost to patch a single computer from \$222 to \$40, representing an annual savings of over \$180,000 for an organization with 1000 workstations.

**Footnotes:**

1. "The Top 10 Requirements for Enterprise Patch and Vulnerability Management" is accessible at [www.patchlink.com](http://www.patchlink.com).



**Figure 2: Elements of A Typical Patch Management Process**

**Research** involves identifying new vulnerabilities and patches that are applicable to the organization. Although straightforward, this task can be time consuming if accomplished manual. An automated approach can save the effort of sifting through a plethora of vendor and relevant security websites, press releases, and email notifications.

**Analysis** begins by establishing the general extent to which a given patch is applicable to the organization – approximately how many systems are affected and what roles/services/applications are they supporting. Inventorying capabilities of an automated solution can facilitate these sub-tasks. This information is then combined with other factors (e.g., severity of the vulnerability, presence of an associated threat, business criticality of affected systems, and availability of other mitigating controls) in the highly manual task of analyzing and deciding whether the given patch should in fact be deployed. Indeed, another unfortunate yet all-too-real consideration that must also be factored in is the potential that any given patch will have negative repercussions on business operations (e.g., by causing system

crashes, or even by introducing additional vulnerabilities). This will often lead to blanket policies, such as “for critical servers, only apply patches associated with critical vulnerabilities”.

**Testing** involves applying each patch (typically individually) to a small subset of each type/build of computer that is affected and then monitoring them for any adverse side effects while the systems and their applications are “exercised”. While the first part of this step can be accelerated by an automated solution, there is minimal opportunity to improve the second part.

**Preparation** starts with the highly manual effort of deciding on the particulars of how to deploy a patch, or more likely, a package of several patches. This entails answering questions such as: Which machines should be excluded? How will reboots be handled? Will the rollout be phased and, if so, how? What are the timing details (e.g., deadlines, maintenance windows)? It also involves collecting the patches themselves, and finally scripting or otherwise configuring the details of the deployment plan into an appropriate tool.

**Deployment** of a patch package and any subsequent troubleshooting that is required can be aided significantly by an automated system, particularly one that is agent-based. In contrast, a manual approach will typically involve directly administering patches to a select subset of machines, as well as a higher failure rate for remote, script-based installations – with both cases requiring a physical visit to the computers in question.

**Monitoring** involves reporting on patch deployment and status (e.g., for compliance purposes) and then validating that all of the patches remain properly installed. Validation should also be repeated on periodic basis, since it is well established that approximately 20% of all systems will become “un-patched” over the course of a year (e.g., due to the installation of old versions of components, such as DLLs, by new patches, applications, or system rebuilds). In any event, both of these tasks can be challenging if done manually, requiring generation of custom signatures, scanning scripts, and reporting mechanisms.

**Table 1: Cost Comparison of Manual and Automated Patch Management**

Variables & Assumptions		
		Notes
<b>Number of computers</b>	1000	
<b>Classes of computers</b>	2	
<b>Applicable patches per year</b>	320	Annual Average for Microsoft *2 to account for other apps & systems
<b>Install rate</b>	75%	One per month plus 2 out-of-phase cycles to account for emergencies
<b>Install cycles (i.e., rollouts)/yr</b>	14	
<b>Hourly rate (\$'s)</b>	50	
<b>Workdays/yr</b>	250	
<b>Install failure rate, manual</b>	15%	Scripts don't work properly, glitches due to custom images, etc.
<b>Install failure rate, automated</b>	1%	
<b>Local install rate, manual</b>	15%	Pre-emptively decide to patch locally
<b>Local install rate, automated</b>	0%	

Patch Management Process					
Task	Task Units (i.e. frequency)	Hours per Task Unit		Annual Labor (Hours)	
		Manual	Automated	Manual	Automated
<b>Research</b>					
Identify available patches	per workday	.5	.17	125	43
<b>Analysis</b>					
Establish scope of applicability	per patch	.5	.17	160	54
Determine whether to install	per patch	.5	.5	160	160
<b>Testing</b>					
Install in test environment		.5	.17	140	5
Establish impact		2	2	56	56
<b>Preparation</b>					
Determine distribution plan	per class/rollout				
Determine distribution plan		1	1	28	28
Compile patches		3	0	84	0
Script/Configure plan detail		9	.25	252	7
<b>Deployment</b>					
Local installs in production	per computer/rollout				
Local installs in production		.5	.5	1050	0
Troubleshoot failures		1	.25	2100	35
<b>Monitoring</b>					
Reporting	per rollout	8	.17	112	2
Validate installation	per month	15	.17	180	2
				<b>Total</b>	<b>4447</b>
					<b>392</b>

Summary of Costs			
	Manual	Automated	Notes
<b>Patch Process</b>	\$222,350	\$19,604	
<b>Patch Management Software</b>	\$0	\$18,000	
<b>Patch Management Hardware</b>	\$0	\$800	annual cost = one time cost divided by 3 years
<b>Patch Management Training</b>	\$0	\$250	annual cost = one time cost divided by 3 years
<b>Patch Management Installation</b>	\$0	\$400	annual cost = one time cost divided by 3 years
<b>Annual Maintenance</b>	\$0	\$480	20% of one-time hardware costs
<b>Total Annual Costs</b>	\$223,350	\$39,534	
<b>Total Annual Cost Savings</b>		\$182,816	

**ABOUT THE AUTHOR**

Mark Bouchard, CISSP, is the founder of Missing Link Security Services, LLC, a consulting firm specializing in information security and risk management strategies. A former META Group analyst, Mark has assessed and projected the business and technology trends pertaining to a wide range of information security topics for nearly 10 years. He is passionate about helping enterprises address their information security challenges. During his career he has assisted hundreds of organizations world-wide with everything from strategic initiatives (e.g., creating 5-year security plans and over-arching security architectures) to tactical decisions involving the justification, selection, acquisition, implementation and operation of their security and privacy solutions.

Dennis Roberson is Regional Director for the Mid-Atlantic for PatchLink Corporation.



PatchLink Corporation  
Scottsdale, AZ 85255  
480.970.1025

[www.patchlink.com](http://www.patchlink.com)