next I.T.

A Patch In Time: Considering Automated Patch Management

Vulnerabilities are followed by patches, followed by exploits, followed by misery. Automated patch management solutions ease the pain and cut costs.

Victor Barra simply didn't have the staff to keep the more than 1,000 Windows servers at Siemens Medical patched. More than once, malware attacks brought down systems and cost the company hundreds of thousands of dollars. Last summer's Blaster worm was the last straw.

"We had to turn the network off at the routers until everything got patched," says Barra, administrator for the medical equipment manufacturer's Windows networks. "It cost the company a ridiculous amount of money.

"Management finally realized we had to put resources into the problem," he adds. For Siemens Medical, that meant investing in an automated patch management solution rather than thousands of man-hours per month. Over the next six months, he plans to use the software to patch 3,500 clients at the company's Malvern, Pa., IT facility as well as 10,000 clients across the country.

The Money Trap

Patching: You're damned if you do and damned if you don't. Some companies, like Siemens Medical, don't have the time and resources to patch thousands of systems, while others commit time and resources they can ill afford. RKA Petroleum, for example, has only about a half-dozen Windows servers and 35 clients to worry about. That was still too much for Jason Hittleman and his staff of two.

"Patching took almost 120 hours a month among the three of us, almost a full-time position" says Hittleman, VP of information systems for the Michigan-based regional petroleum distributor. "Employees had to stop work or work at another station. It was hell." He now spends about an hour patching each month, using an automated patch management software.

With an ROI that's impossible to ignore, automated patching is a cost-effective alternative to the drain of manual patching -- and the risks of not patching. Automated patch solutions can fit nicely into your vulnerability management program, drastically reducing costs and narrowing the window in which your mission-critical systems are exposed.

But that doesn't mean that automated patching -- or any patching at all -- is necessary or desirable in every network environment. It all depends on the value of your protected assets, the threat level, the presence of other mitigating factors and the required effort and resources.

For one thing, patching isn't always as cost-effective as it may seem. Time is money, and patching takes time. So, it's important to measure the cost of patching and not patching against the level of risk, and then determine when and what to patch.

To calculate costs, use the following equation:

(Hours x Rate x Systems) + (Patch Failure% x (Hours x Rate x Systems)) = Cost to Patch

So, if it takes an army of $70/hour technicians one hour to patch a system, and there are 2,000 systems, the cost is $140,000. If you estimate that 5 percent of the patches fail, and figure an average of two hours of recovery time (which includes help desk and IT support activities), that's 100 systems at $140 each -- another $14,000.

Using an automated system with multiple threads reduces the 2,000 man-hours to a few automated hours -- it's easy to see why the ROI is so high.

Consider, too, that time isn't necessarily a constant -- several factors have an impact. Heterogeneity of systems -- variations in platforms, configurations and deployed applications -- increases the chances that patches will fail, because nonstandard systems are hard to test. Sheer numbers have a significant impact on the amount of time it will take to patch all systems and contribute to the likelihood that a percentage of systems will be missed.

And highly decentralized environments will take longer to patch as notifications are sent to affected business unit managers and consensus is built across the organization on how to proceed.

The decision about when to patch is important, but often overlooked.1 You should patch at the earliest point in time where the cost to patch is less than or equal to the cost not to patch. For example, the risk to unpatched systems -- and the potential cost in downtime and recovery -- increases dramatically once an exploit is publicly available.

The other side of the equation is weighing the risk of leaving your systems unpatched. Put another way, what's the cost of not doing anything? This is a far more complex determination, incorporating the cost of recovery for stricken systems and the value of the system at risk adjusted for the likelihood that it will be compromised.

The value of the system, along with its corresponding risk of loss, is an element that continues to elude security professionals. Regardless, it's important to at least "gut check" the potential for loss. There are six elements of loss for any computing asset:

- Lost productivity for the end user
- Lost productivity for IT support personnel
- Loss of revenue (direct)
- Legal/regulatory costs
- Intellectual property losses
- Loss of stored assets (financial)

Each of these elements helps you quantify the potential loss of an asset, whether the breach compromises confidentiality, data integrity/availability or system availability. With worms and viruses on desktops, for example, the losses are typically in the productivity of the desktop itself, but are exacerbated if the desktop infects the rest of the network.

Next, consider the factors that determine the risk of an exploited vulnerability on the computing environment:

- The impact of the vulnerability is about damage or the payload of the attack. Worms and viruses can delete files, install Trojan software, provide a root shell, and participate in a distributed DDoS attack.
- Likelihood corresponds to the exposure of a system to exploit code.
- Criticality is the functional value of the system to an organization.

Do The Math

Now that we know both sides of the equation, let's crunch some numbers. For this scenario, we'll assume a vulnerability has just been discovered that causes a DoS attack against client PCs. The payload is light, but it still requires a full system install to recover.

Taking our earlier example of 2,000 systems, we estimated patching would cost $154,000.

On the other side of the equation is the cost of not patching. Let's again figure two hours of recovery time at $70 per hour, and estimate that initially only 10 percent of the systems are at risk. This gives us a cost of $28,000 for 200 systems. In addition, we estimate that each of the 200 at-risk systems has a productivity value of $200 (lost availability to end users). This adds another $40,000, for a total of $68,000.

So, we have a cost to patch of $154,000 and the cost not to patch (at risk) of $68,000. In this scenario, it makes no sense to patch.

Adding an automated patch management solution to the equation dramatically changes the numbers. The annualized cost of the software, say $50,000, replaces the $140,000 in labor. Now spread that $50,000 across many patches. So, 10 patches in a year effectively reduces the cost to patch of each incident to $5,000, plus the $14,000 to recover from patch failures. Compare $19,000 rather than $154,000 against the $68,000 at risk. Are you going to patch? You bet.

The numbers are even more dramatic once exploit code is released. Assuming the risk of attack jumped from 10 percent to 50 percent, the cost not to patch shoots up fivefold to $340,000. If you're under the gun, you have to patch, and automation makes it a lot quicker and a lot cheaper.

What to Look For

Once a patching strategy is in place, execution becomes paramount. For patches, the clock starts ticking as soon as the vulnerability is found. With days and months of lag time between discovery, notification and patch distribution, it's already too late. Automation becomes the only alternative that makes sense, particularly with the ROI associated with productivity gains.

There are a number of solutions for automating patch management. Microsoft and many AV vendors have their own patching services for users and enterprises, but they generally don't match up to commercial automation solutions.

When evaluating a solution that automates patch management, consider the following elements:

Platform Coverage:

Patching systems has always been the purview of the respective platform owners. With security becoming paramount in the enterprise, however, the need to centrally coordinate security patching has emerged. Most vendors support Microsoft products, but you should carefully evaluate the desktop and server OSes and applications that are covered.

Research Depth:

Some vendors provide the value in the distribution process, while others add more research depth and perform independent testing. Some provide more information than others, such as a fuller description of vulnerabilities and the files that are affected by the patch.

Workflow:

The patching process typically involves a team of staffers working on related activities. Security researchers will conduct initial investigations and disseminate information; an architecture group may evaluate the patch for applicability; testing personnel will evaluate the patch's "fit" into any standard builds; system and application administrators will apply the patches; and management will review the results. Each step in the process must be tracked and coordinated to keep the production line moving. Current products have limited capability, but expect more work in this area to come.

Controlled Rollout:

A solution should be able to group systems and applications based on attributes like geographic location, functional use in the enterprise, exposure ratings (relative level of risk), or other factors, so that a rollout can match organizational requirements. Siemens Medical, for example, delivers controlled rollouts to specific business/ departmental units during a one-hour maintenance window at 5 a.m. each workday.

Most solutions now offer some sort of grouping capability.

Rollback:

Rollback is the feature you never want to use and always want to have.
The ability to roll back a patch provides comfort to those starting down the path of automation -- to know that any initial hiccups can be reversed.

Automation saves time and money, but you have to relinquish some of the control of manual patching. No software is perfect, and you need the assurance that you can undo the effects of a patch that's applied to the wrong system or causes problems because of inadequate testing, or a misconfigured or nonstandard system.

Validation:

Ultimately, an automated solution validates that patches have been installed across the enterprise. A patch installation may miss a system, for example, because a server is offline for maintenance or a reboot, or a user's laptop is unplugged. Or, a patch may fail because of some system issue -- such as the interruption of the installation. Some solutions will record misses or failures and automatically retry the next time a system connects.

Call us at

1 866-388-6398

or you can e-mail me at

sales@next-it.net