# Leaving the administrator behind

**Wes Miller**

More than three years ago, I began the process of taking my user account on my primary Windows system from local administrator account to local user account. I had worked at Microsoft for more than seven years, always running as a fully privileged administrator. Sure, it was convenient – but that scary lack of security highlights the amazing luck that I (and so many of us) have had, that in running as an administrator every day I haven't caused more damage to more systems, more often.

I often wish that there was a way to get a good statistic on this, but both my gut and the industry tell me that too many organisations – and too many IT pros themselves – are running as local administrators today. At Winternals, when I switched to running as a user, my intention was to learn how hard it was (as a "prosumer") and to see how our product, Winternals Protection Manager, could help in an average organisation. Given that most organisations were, and still are, running with a good percentage of users as administrators, our goal was to enable administrators to run as users, but to make the transition – or at least the pain points – as minimal as possible. Regardless of the technology you use, it isn't easy to move your organisation from one where users are administrators to one where they are users, but it is the single most effective way to reduce the attack surface within your organisation. Think of it as an intra-system firewall, because that's really what it is.

## How did we get here?

That most users act as administrators is rooted in Windows history. With the first versions of Windows, before Windows NT® 3.1, every interactive user was as empowered as the next – functionally, there was no security. In the home, this wasn't terrible; it meant that all software installed the same way. The assumption was that the user owned the computer and that all software was installed for all users of that computer.

When Windows NT first appeared, it certainly didn't immediately own the enterprise (let alone the consumer) market. And because of the Win32® compatibility between 32-bit Windows and Windows NT, most application vendors didn't rebuild their applications just for the sake of the security infrastructure of Windows NT. In fact, it really wasn't until Windows 2000 that many consumer-oriented independent software vendor (ISVs) started paying attention to Windows NT. It was Windows XP, of course, that forced the issue as it ended the 9x family of Windows.

But still, applications rolled on, assuming that every user on the system had access to write to Program Files (users don't), and HKEY_LOCAL_MACHINE (HKLM) in the registry (users don't), and HKEY_CLASSES_ROOT (users don't). Games are often among the worst offenders in assuming access; see Matt Clapham's article on this topic at http://technetmagazine.com/issues/2007/02/Gaming.

This is problematic because most cross-system apps store their files and registry settings in those locations and you need to be able to read and write to those locations in order to be able to install them. Unfortunately, some apps then insist on writing to those keys after installation. For example, my daughter has a game that is Flash-based. It attempts to install a custom player *every time you run it* – meaning that when my daughter runs as a user, not an administrator, the application fails to start, with a fatal error. While this is extreme, and it is a consumer application, the reality is that many non-consumer applications still don't play well in the world of non-administrative users. In fact, if you follow up on my challenge (see the sidebar, "My challenge to you: feel the pain"), you'll discover just how much Windows itself isn't tolerant of running as a user.

If you take a look at **Figure 1**, you'll see what running IPConfig /release as a user looks like on Windows XP. If you compare that with **Figure 2**, you'll see that the same command under Windows Vista is not that much better, but at least you know why the command is failing. Note that the networking tools as a whole have been improved to allow users to refresh their IP addresses. Similarly, trying to run Computer Management (compmgmt.msc) as a user under either version lets you perform a limited number of tasks – but generally results in frustrating dead-ends, as **Figure 3** shows. While Windows Vista doesn't initially enable many of the tools in Computer Management, it does present clearer access denied messages.

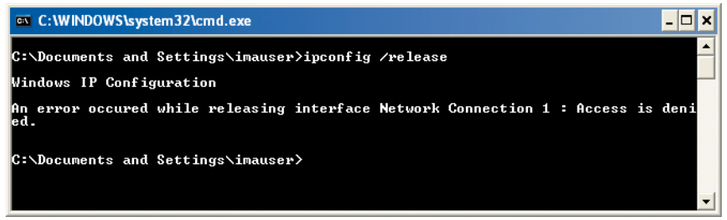


Figure 1 **Running as a user under Windows XP**


Figure 2 **Running as a user under Windows Vista**

## Why It matters

So why should you care? Because we, as IT professionals, should begin forcing applications to adjust to least-privileged users, instead of vice versa where applications assume the interactive user owns the system.

Unfortunately, the same policies that allow administrators to write to registry keys also grant any malware run in their user context full access to anything they have not been explicitly denied via access control lists (ACLs). In the world of UNIX, people follow the rule regarding not running as root (the functional equivalent of the Windows Administrator account), mostly because the ecosystem of software that pushes the boundaries of the security model is tiny to nonexistent.

Still, the best thing you can do is to follow that same wisdom and only run as an administrator when it is explicitly required – or better yet, only run individual applications as an administrator. By doing so, you raise that intra-system firewall I mentioned earlier. Then, when malware or spyware attempts to do something it shouldn't, it fails – because it can't write to the registry or file system locations it needs in order to really infect your system (such as installing a service or driver, or installing for all users). In addition, doing so allows anti-malware software

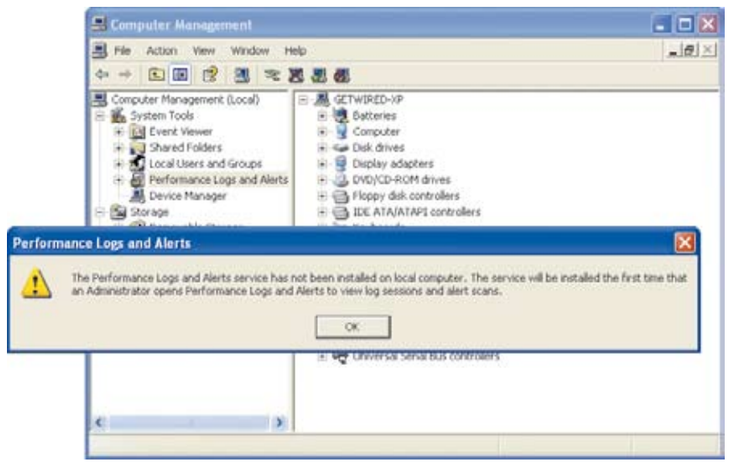


Figure 3 **Misleading message after running compmgmt.msc as a user on Windows XP**

## My challenge to you: feel the pain

If you haven't yet begun thinking about moving your administrators to users, you should. And I'd encourage you to start by experimenting with it yourself. Not on a secondary machine – that's cheating. Try it on your primary system, the one you use every day. I'd also encourage you to even try it without using User Account Control (UAC) if you are running Windows Vista. When you begin evangelising an effort within your organisation to change something, it's a good idea to be a practitioner of it before you preach. I think you'll find that running as a non-admin isn't really that difficult – and that with the added security of doing so, you'll really change your organisation's attack surface.

to contain malware that it recognises, without risking the entire system first.

Note, however, that users aren't impervious to attack. Though this class of malware doesn't exist widely yet, there is the potential for malware to successfully infect an individual user's context or destroy his data. But the attack vector posed by such software is limited. As a result, the same thing that keeps malware instances low on Linux or the Macintosh (namely the lower number of potential victims) can help ensure the general safety of your end users – and yourself – today.

## Wither power users?

When we were developing Protection Manager, one of the comments we heard from customers was, "We're running Windows XP with all of our users as power users, not administrators, so we're secure." The reality, though, is that power users are just a few steps away from administrators. There are several potential holes that would, with a little work, allow a power user on Windows XP to become an administrator. In fact, the power users group was eliminated on Windows Vista and Windows Server 2008; only a system upgraded from an earlier version of Windows will have a power users group. All in all, you should always avoid using the power users group, even if you are using Windows XP.

## Lossy permissions

In my column on Windows thin clients (http://technetmagazine.com/issues/2008/03/DesktopFiles), I spoke recently against the prospect of thinning out Windows XP to save space. In the same vein, there is a common practice to enable administrators transitioning to users that you will need to consider, but do so carefully. That is the practice of adjusting ACLs on the registry and file system so that users can write to locations they are not normally able to – thus enabling problematic applications.

Obviously, the best choice is to get an updated version of an application that doesn't require such a change, but that's not always possible. If you must change permissions (that is, drop them), proceed very carefully. Remember that the firewall between a user and an administrator is defined largely by registry and file system permissions. Opening them up lowers your protection and potentially widens the attack surface posed by malware – so please proceed judiciously.

## What about UAC?

No discussion of transitioning users from administrators to users is complete without addressing User Account Control (UAC) in Windows Vista. UAC, like similar functionality on the Mac OS X, lets you run "as an administrator" without putting yourself at so much risk.

How does this work? Take a look in **Figure 4** at what Process Explorer is showing about cmd.exe. The instance of cmd.exe on the right was started without elevation, with me running as an administrator. As a result, even though the user on the right is identical to the one on the left (where the cmd.exe was started with elevation), the application itself does not contain the necessary privileges and tokens for it (and the user running that instance) to perform any tasks that require administrative rights. UAC works by lowering the attack surface *within a user's*

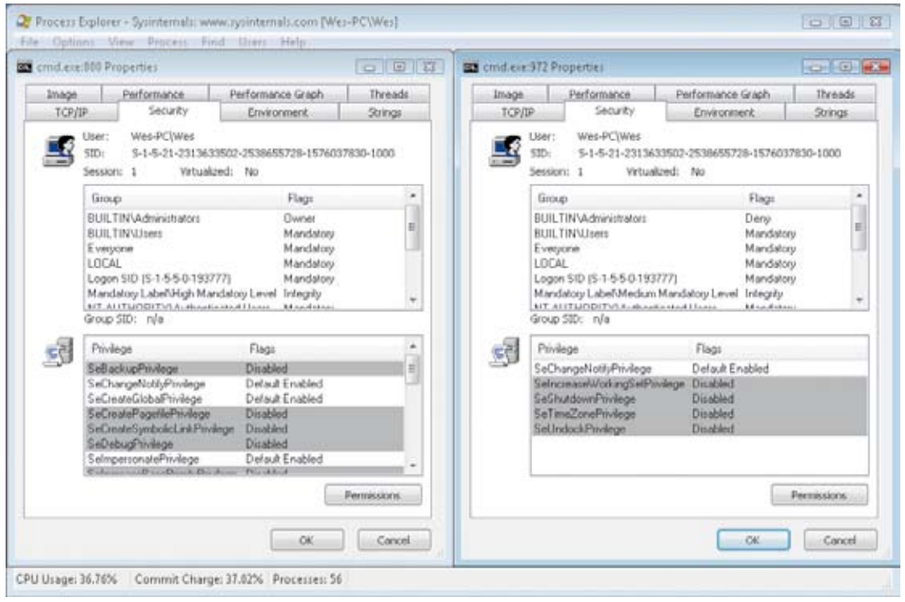


Figure 4 **Two instances of cmd.exe with different privileges**


Figure 5 **Shields in Windows Vista indicate the need for elevation**

*interactive context*. The only problem is that something has to tell Windows that this task requires administrative privileges and that the user is OK with allowing the elevation required to complete this task.

The small shields in Windows Vista show you which tasks require elevation (see **Figure 5**). These tasks require elevation every time you run them – and this is one of the sore points the press has chosen to highlight with Windows Vista. The alternative, letting the credentials be more "sticky," poses a potential security threat that could be used to more easily exploit the system.

If UAC is enabled and your users are running simply as users, they will be prompted for a set of administrative credentials when an application requires administrative privileges. As when using runas or psexec, the app runs in the context of the user you launched it as – unlike what happens under UAC when running as an administrator, when the tasks run in your context but with elevated privileges.

## Running Windows Vista as a user?

My personal preference when running Windows Vista is to actually run as a user, not an administrator with UAC, because I believe that in the average enterprise, this is still the best idea. After all, your users have full control over their systems and you may have decreased the window of opportunity for malware.

Moreover, if you are intending to manage your users with Group Policy, antivirus, antimalware or other software and you want central control over whether these tasks actually get enforced or completed, ensuring your end users are not administrators is a critical step. If your users are administrators, they can stop services, add or remove drivers, and more. Of course, a crafty end user running as a user can use Windows PE to bypass some security hurdles. BitLocker can make that more difficult, but again, remember that end users with physical access can do whatever they want to their machine, given enough time, knowledge and dedication.

Running Windows Vista as a user isn't that much different from running Windows XP as a user. I use the same tools – PSExec, RunAs (and now UAC) to run tasks as an administrator. The nice thing is that quite a few tasks in Windows XP that used to require administrative privileges no longer do. For example, a Windows Vista user can install a local printer. (Network printers could be installed by users in Windows XP, but installing a local printer required administrative privilege.) In Windows Vista, as long as the user is physically at the machine and the printer driver is in the driver store, a user can install a printer and manage print jobs on it (see http://www.microsoft.com/uk/windowsvistaprintersetup for more information). This functionality is disabled in Windows Server 2008.

Of course, people make fun of the clock functionality (or lack thereof) when running as a user in Windows XP. Try double-clicking on the clock as a user (something people often do to see what the date is, whether or not it was designed to work that way) and you'll get the error shown in **Figure 6**. Not very friendly. In Windows XP, you can modify policy so that users can do this, but in Windows Vista it was changed to just work that way. So all in all, running as a user – whether you're using UAC or running as a formal user and elevating as another user – is generally more palatable on Windows Vista than it was on Windows XP.

## Remember the limits

Remember that moving your users to be non-admins isn't a panacea. Dedicated end users are still physically located on their own PCs, and they can work pretty diligently to exploit their own systems, especially if the policy or user privileges are inconveniencing them or preventing them from getting their work done.

If your users are running as administrators, it doesn't take much work to bypass any Group Policy in place. Of course, with a little more work, users can boot to Windows PE and modify permissions that they wouldn't normally have privileges to – though if you use BitLocker or other drive/volume encryption, you can make that impossible, or at least more difficult.

The most important thing you can do if your organisation has not yet

begun transitioning end users to run as users is to familiarise yourself with the reasons why you and the organisation should spend time, money and effort to move away from having users running as administrators.

Sure, legacy applications can be hard to let go of, but if you have an application that simply cannot be run as a user, it's just a bad idea to hold on to it at the expense of your organisation's security. You should consider virtualising the application – literally moving it into a virtual machine where the user is indeed an administrator. This lets the application be used as needed, but still allows you to secure the rest of the system by moving the administrators to users.

Note that through this entire column, I have not used the word "lockdown" or any derivative of it. Many people consider moving administrators to users a part of a task often described using that word. Perhaps it's my psychology background or my current marketing world, but I think it's important not to use words that make your end users feel that their privileges are being taken away (even though, at a semantics level, they are).


Figure 6 **In Windows XP, non-admins couldn't change the time**

Instead, focus on the security benefits to the organisation and make sure you have a good plan for edge cases where a specific user cannot run as a user or there is a task that requires admin privileges. Whether you use something manual like my Run.vbs script (which you'll find at http://technetmagazine.com/issues/2007/03/DesktopFiles) or a commercial solution to help you make the transition (that lets you hide the details from your end users and makes things "just work"), it's important to start heading down the non-admin road as soon as you can. Frequent *TechNet Magazine* contributor Aaron Margosis is *the* evangelist when it comes to running as a non-administrator. If you aren't familiar with his blog, you should be – it's the best place to go for in-depth information on this topic (see http://blogs.msdn.com/aaron_margosis). ■

**Wes Miller** *is a senior technical product manager at CoreTrace (www.CoreTrace.com) in Austin, Texas. Previously, he worked at Winternals Software and as a program manager at Microsoft. Wes can be reached at technet@getwired.com*

**Utility spotlight**

# Offline Virtual Machine Servicing Tool

**Peter Skjøtt Larsen and Suveen Kumar Reddy Vuppala**

Virtualising a computer workload simply means using a virtual machine (VM) to untether the workload from the underlying hardware. Modern IT departments find VMs useful in many situations, including:

**Shifting workloads** You can easily use VMs to increase or decrease the workload bandwidth of multiple setups, as your usage demand changes, without having to make a corresponding hardware change.

**Developing and testing applications** It's possible to create multiple VMs that represent each configuration an application is supposed to support without needing dedicated hardware.

**Software upgrades** You can use VMs in order to bring the new version of a software package online as you take the previous version offline – all on the same hardware.

**Software distribution** VMs can be used as a unit of distribution for a line-of-business application that has a consistent combination of software in a pre-tested package.

## Security nightmare

One of the advantages of using VMs is that you are able to store them in an offline state as VM images. Then, when you need them, you can "wake" these VMs and deploy them much more quickly than you could deploy the equivalent hardware.

Keeping an increasing number of computing environments waiting offline presents a maintenance challenge, however. Many software update mechanisms rely on systems to be online in order to check for updates or to receive updates automatically. When a VM is not online, it is just a file sitting in a computer, so it cannot interact with any update mechanism. A VM that has been brought online after being offline for a few months thus might become a threat to the network, or the network could threaten it.

It is not simply a matter of missing OS updates. Outdated applications or virus profiles can render the VM vulnerable or out of compliance with company standards.

To help customers address the challenge of keeping offline VMs up-to-date, the Microsoft Solution Accelerator team has created the Offline Virtual Machine Servicing Tool.

This tool works with System Center Configuration Manager (SCCM) 2007, Windows Server Update Services (WSUS) 3.0 and System Center Virtual Machine Manager (VMM) 2007 to orchestrate the updating of stored VMs. **Figure 1** shows a conceptual rendering of the tool and how it connects to various external components.

To make VMs available for updates, the tool uses VMM to temporarily deploy them to maintenance hosts. Because a maintenance host configuration typically has the necessary CPU and memory to run multiple VMs at the same time, the tool can manage VMs in batches.

As soon as the VMs are active on the maintenance hosts, either SCCM or WSUS can supply them with the necessary updates. After the updates have been applied, the tool uses VMM to return the VMs to their offline state. (Note that the tool only supports VMs that are managed by VMM.)

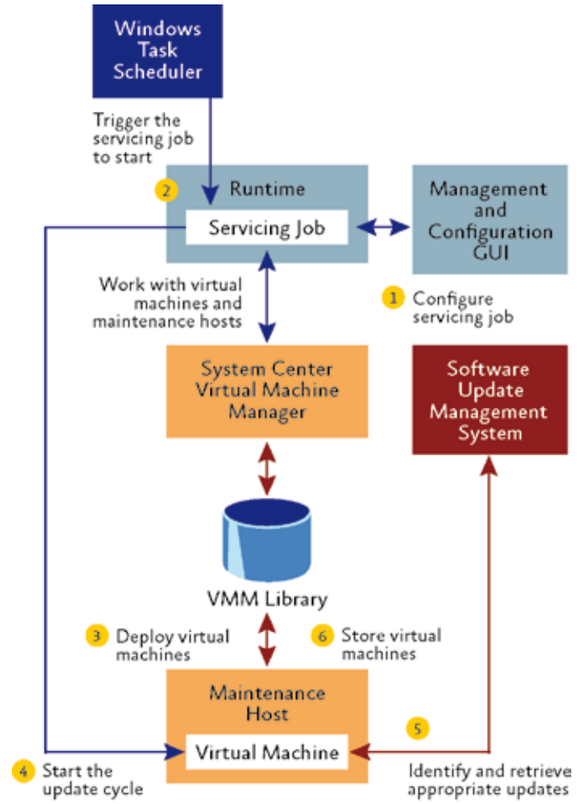


Figure 1 **How the Offline Virtual Machine Servicing Tool works**

## Under the hood

The Offline Virtual Machine Servicing Tool uses Windows Workflow Foundation (WF) to orchestrate the process of updating a VM. The process has a number of decision points, beginning with choosing the appropriate update management system, picking the next available maintenance host appropriate for the VM, ensuring that the update occurred, and, finally, dealing with exceptions.

Using a Windows WF-based solution gave the development team great flexibility to change and evolve the process. It also offers users a robust solution that can be tailored to meet specific needs; at critical junctures in the process, built-in pre- and post-workflow steps provide opportunities for customisation.

The tool uses Windows PowerShell to implement individual tasks below the workflow level, which ties in nicely with the Windows PowerShell API offered by VMM. And the Microsoft .NET Framework-based UI looks and feels like System Center products, so new users should feel at home.

## Servicing infrastructure

One of the basic principles of the servicing infrastructure is to configure network security to ensure that VMs don't get damaged while the update is happening. In version 1.0 of the tool, this is accomplished using a virtual private network (VPN) to which VMM and the appropriate update system (WSUS or SCCM) connects. The most appropriate infrastructure for a VMM library is a Fibre Channel-connected storage area network (SAN), which provides the means for fast transfer of VM images to the maintenance hosts.

All the VMs must be members of the same domain, one that uses Active Directory and DNS. Separate servers can be dedicated to VMM, WSUS, SCCM and the VMM library, but

combinations of virtual servers can also be used for smaller environments. Needless to say, the maintenance hosts must be physical servers.

## Using the tool

After you have set up the servicing infrastructure, you need to check that certain settings are correct before the tool can start. Make sure that VMM is managing all the appropriate VMs, that each VM has the appropriate update client installed, and that the necessary update packages are configured in WSUS or SCCM. Ensuring that groups of maintenance hosts are configured in VMM is optional.

When you are ready, start the Offline Virtual Machine Servicing Tool, which has a number of configuration steps of its own. You will have to designate the VMM server and the appropriate WSUS or SCCM server, and then specify which group of maintenance hosts to use (if the maintenance hosts are grouped) and which maintenance hosts from that group to use (see **Figure 2**). You may want to configure groups of VMs to be managed, but this is optional.

After you have configured the tool, you create the servicing jobs. A servicing job contains all of the information the tool uses to manage specific VMs, including whether to use WSUS or SCCM for updates; locations of the VMM server and the WSUS or SCCM server; identities of the VMs to be managed; type (and identity, as appropriate) of network to use for the process; identities of the maintenance hosts to use; account credentials needed to access the VMs, the VMM server and the WSUS or SCCM server; and, finally, the schedule for running the servicing job (run immediately or at a specific date and time).

If you specify a date and time for the servicing job, Windows Task Scheduler determines when to start it. As the servicing job runs, the Offline Virtual Machine Servicing Tool follows this sequence for each VM:

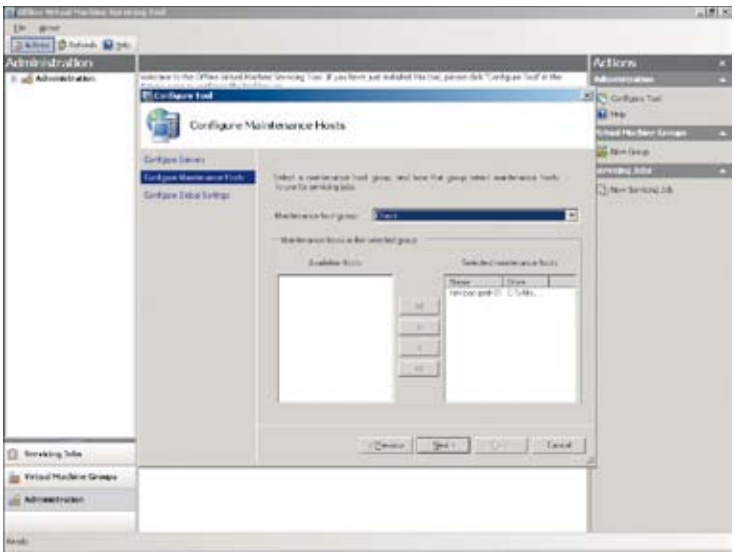


Figure 2 **Configuring the Offline Virtual Machine Servicing Tool**

- Select the next VM from the VMM library.
- Query VMM for the most appropriate maintenance host.
- Deploy the VM onto the maintenance host.
- Ensure that the VM connects to the correct network.
- Start the VM.
- Make sure the appropriate update client is installed on the VM.
- Trigger the update process.
- Wait for the update process to complete.
- Shut down the VM.
- Store the VM back in the VMM library.

The time it takes to update a library of VMs will vary greatly, depending on the number and capacity of the maintenance hosts, the access speed of the VMM library storage, and the nature of the updates.

## Coming up

The current version (1.0) of the Offline Virtual Machine Servicing Tool does not support network access protection (NAP), which is a really attractive way to protect VMs from the network. In addition to this, it does not support the Hyper-V technology of Windows Server 2008 or the use of Windows Server 2008 as a client OS.

Newer versions of SCCM, WSUS, and VMM will be available soon, and version 2.0 of the Offline Virtual Machine Servicing Tool will provide support for them as well as for Hyper-V and the use of Windows Server 2008 on the client. The tool is also going to support NAP for network isolation. You can download the Offline Virtual Machine Servicing Tool at http://technet.microsoft.com/cc501231. ■

For more information on Virtual Machine Manager, visit the VMM TechCenter at:
*http://www.microsoft.com/systemcentervmm*

**Peter Skjøtt Larsen** *is a senior product manager at Microsoft. Before joining Microsoft, Peter was involved in both architecture and development of telecom operational software systems and standardisation and development of wireless services.*

**Suveen Kumar Reddy Vuppala** *is a senior software development engineer at Microsoft. He was previously involved in designing and developing the real-time monitoring tools and deployment solutions for Microsoft for the past seven years.*