MAX RemoteManagement™

MAX INSIGHTS

MAXfocus Whitepaper

Windows Server 2003 End of Life: Defusing the ticking bomb

We are Max

MAXfocus™
From LogicNow

Introduction

Solving the Windows Server 2003 end-of-life problem is no easy task, particularly when it comes to the migration of poorly or unsupported, in-house or vendor-developed applications. A lot of MSPs and in-house IT shops have already endured significant grief migrating SMB business applications to Windows Server 2008 and Windows Server 2012.

Unless the vendors of the original business application can provide a clear upgrade, which allows for installation on the new server platform as well as an import of the old database into the new database, you are likely to be confronted with the arduous, and unpredictable, task of installing the software (if you can find the software and its license keys), copying over the database, and then "seeing what happens." Once in a while you score, and not only does the software install, but the database moves over without a hitch, and all is well. The reality is that most of the time it tanks.

Permissions, missing registry keys, windows firewall, and a cornucopia of incompatibilities lead to one central issue: the 32-Bit/64-Bit Problem.

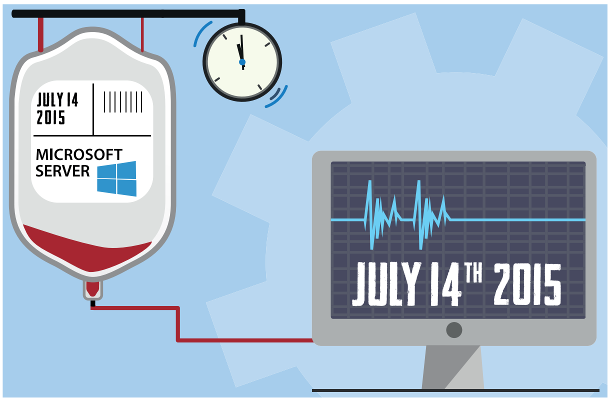
Windows Server 2008 and Windows Server 2008 R2 are the last Microsoft 32-Bit Operating Systems. Windows Server 2012 will only be available in a 64-bit version. Windows Server 2003 had a later 64-Bit version and if you're lucky enough to be on that version your upgrade experience will be way more pleasant. For those with working applications on the 32-Bit version of Windows Server 2003 there may be a "new fresh hell" waiting for you.

The problem arises from old applications that may contain or use 16-Bit Dynamic Link Libraries (DLLs). I've seen a number of these applications still using programs like Obview.exe and a host of other 16-Bit DLLs. The presence of a config.ini file potentially betrays the presence of an "ancient" legacy application and the challenge that awaits you. While it's possible some of this code will run fine on a modern OS, the more likely scenario is that it will "partially" run, and the 16-Bit functions will simply lurk in the background until they are specifically called by the alleged "32-Bit" program, at which point they will fail.

The situation is further complicated when applications are built on older versions of Internet Information Service (IIS) or (heaven, forbid) Microsoft Front Page Server Extensions, thus possibly requiring a legacy version of Internet Explorer that is not supported (and may not run) on Windows 7 or Windows 8.1. I even know of one legacy application that requires Netscape Navigator!

So, given all this potential for disaster, what is the best approach to take?

If you enjoy spending an extraordinary amount of time trying to move an old .asp application on an ancient version of IIS to a new version, then good for you. There is, however, an easier way:



Step 1: Virtualize the Windows 2003 Server

Not surprisingly, the hardware your Windows Server 2003 machine is running on is probably long overdue for retirement. If it came pre-installed on the server when you bought it, then the likelihood is that it is somewhere between five and 11 years old. On the other hand, if you upgraded from a Windows Server 2K install, you probably have some truly ancient boxes. Either way, your hardware is more than likely out of warranty, its RAID card battery is dead (or at least dying), it may have a dead drive in it, and it may even look like this:

So, the first step to migration nirvana is moving your 2003 Server off this ancient hardware and into a Hypervisor or VMware Virtual Machine environment running on a robust 64-Bit OS. Both Microsoft and VMware have solid tools and strategies for moving from P2V, Physical to Virtual. VMware Center Converter and Microsoft System Center Virtual Machine Manager (SCVMM) are the most popular options. In the case where it's not just applications that are the potential issue, and the Windows Server 2003 is also the Domain Controller, DNS or DHCP server, you can migrate these services to a more modern VM or Hypervisor, or even have the physical server do these roles.

A little bit of tweaking and all of a sudden your turning your old hardware off and running Windows Server 2003 in a virtualized environment. Perhaps your old hardware can be someone else's Disaster Recover/Business Continuity Plan. Also, keep in mind that you won't need to renew those backup software licenses anymore as you're running virtually and can use snapshots, as well as the native backup capability of the physical server itself.

As you make changes to your "test" or development copy of the now virtualized Windows 2003 server, clear documentation is key. Hardening and un-installing applications and services on the server may have unanticipated consequences. It's prudent to be able to identify the step that caused instability and research a work around. Eventually you will have a list of steps to take that you can apply to the production Windows 2003 server, with some assurance that your changes will not bring the production server to a halt.

The big bonus of virtualizing your old Windows 2003 server is that you can copy the VM or Hypervisor onto another machine, or a robust laptop. This means you now have a development 2003 server to test your production server, without the risk of pulling it down.

Step 2: Secure the Windows 2003 Server

Chances are you're already running antivirus on your server. However, we still haven't solved the issues of whether you're Windows Server 2003 server has to be on the Internet, accept legitimate business network traffic or provide services to the business.

The first step to securing your old server is removing and disabling as many services or applications on the machine as you can. You need to move AD, DNS, DHCP (Including all the FSMO roles) to another server (or dedicated appliance); Print Queues, Remote Access Server (VPN), RDP, FTP and any applications can behave perfectly fine on another more secure platform. Meanwhile, unless they're needed by your problem application(s), programs such as Adobe Reader, Java, Flash, QuickTime and Shockwave need to be evicted from your new 2003 Virtual Server environment.

Reducing the services and applications installed on the Windows 2003 server, reduces the attack surface and may improve application performance. If you have managed to move File and Print services to another machine and file shares are no longer required on the Windows 2003 server, then try shutting down the server service. You may find a big boost in performance. It's impossible to be infected by an exploit, which targets Java, if Java is no longer installed on the server.

A few minor firewall adjustments, and your Internet-facing services are now safely migrated to your more secure, modern operating system.

So, with core network services moved off the 2003 Server it's time to ask yourself a couple of questions.

What remains on the server that's critical?
And, does any of this genuinely need access to the Internet?

Step 2: Secure the Windows 2003 Server

If it's just an in-house app that has to live on the 2003 Server, can you secure it with a firewall rule and prohibit it from communicating to and from the Internet? If not, and the server has a legitimate need to communicate to the Internet, you now have some challenges.

Of course, you already should have a firewall rule specifically identifying the source and destination for services like Electronic Data Interface or API connections - you don't want the world probing these sensitive connections - especially if they interface or hook into the old IIS installation. Perhaps if they are really sensitive and "wide open" you need to secure them at the network layer with VPN, SSL, IPSEC or MPLS technologies. A robust Geo-IP filtering capability is also advised - there's no need for Russia to be probing your business API.

For those running business applications that integrated with an older version of MS Exchange (like an older Razor's Edge, or other legacy ERP or CRM applications and the dreaded Winfax Pro), an upgrade path maybe murky and the business processes may be non-flexible. If this is the case, you're going to have to face the realities that this server needs to be on the Internet to send and receive email and possibly provide that email to BYOD or some other devices attached to the old version of IIS.

But don't worry, there are more things that can be done.

You have patched and updated the 2003 Server (maybe with the last patches and updates ever to be issued). You have removed as many of the services as you can and migrated them over to newer, more secure platforms. You have landed or upgraded to MAXfocus Managed Anti-Virus. You have deployed MAXfocus Web Protection to prevent malware that may infect the VM from reaching Command and Control servers. You have tested and hardened the permissions in your test environment and the suggestions that seemed to work fine have moved into production. If you still have dependent 3rd Party applications that need to still be installed keeping MAXfocus Patch Management on the server to keep them up-to-date is a great idea.

Now, you can extend the defensive perimeter to the Cloud - specifically when it comes to email. Putting a cloud-based email scanning service such as MAXfocus Mail Protection in front of your legacy exchange server and writing a firewall rule to only accept email from the Cloud service will do a great job in preventing "Windows 2003 exploit" laden emails from causing grief to the server and your users. Conversely, only sending emails from your server to the Cloud service will give you the heads up if something strange happens; for example suddenly having 10,000 emails erupting from your business.

Step 3 – Backup the Windows 2003 Server Virtual Machine to the Cloud

When it comes to fending off the worst the Internet has to offer in the middle of 2015, all your hard work in Steps 2 and 3 will be for naught, if you can't restore business services quickly.

MAX Backup is a great solution for hosted backup stored safely in the cloud - out of reach of potential ransom ware attacks.

It stands to reason that if you could not move away from Windows Server 2003 for business reasons and even if you have done everything you can to reduce the attack service, you're still exposed and that means you're going to get attacked. The possibility of a successful attack is hard to calculate and depends a lot on how much effort you put in to securing the server.

The key advantage of virtual machines is the portability, backup and ease of migration they offer - they are "just files" after all. It's easy and fast to restore, rollback, or copy a Hypervisor or VM, it's not so easy to do that on a physical machine that may feature an ancient tape backup as the primary restoration mechanism.

With the success of CryptoLocker attacks, it seems reasonable to expect this vector of attack on exposed Server 2003 infrastructure. Maintaining an offsite, encrypted, cloud-based backup of the VM or Hypervisor is not only a good idea, it's a necessity, should you find your VM or Hypervisor platform demanding a ransom.

> *"I am prepared for the worst but hope for the best"*
>
> - Benjamin Disraeli

MAXfocus enables you to offer your customers a robust security service, to find out more give us a call, or visit www.maxfocus.com for more information.

USA, Canada, Central and South America
4309 Emperor Blvd, Suite 400, Durham, NC 27703. USA

Europe and United Kingdom
Vision Building, Greenmarket, Dundee, DD1 4QB, UK

Australia and New Zealand
2/148 Greenhill Road, Parkside, SA 5063

www.maxfocus.com/contact

Disclaimer

The information and content in this document is provided for informational purposes only and is provided "as is" with no warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and non-infringement. LogicNow is not liable for any damages, including any consequential damages, of any kind that may result from the use of this document. The information is obtained from publicly available sources. Though reasonable effort has been made to ensure the accuracy of the data provided, LogicNow makes no claim, promise or guarantee about the completeness, accuracy, recency or adequacy of information and is not responsible for misprints, out-of-date information, or errors. LogicNow makes no warranty, express or implied, and assumes no legal liability or responsibility for the accuracy or completeness of any information contained in this document.