Security Patches Management on a Windows Infrastructure: from Technical Solutions to Real World Implementations

Patrick Chambet (Edelweb) and Eric Larcher (Accor Services)

Introduction

Security patches management is one of the most boring tasks for corporate administration and security teams. However, it's nowadays quite impossible to shirk from this kind of job.

Indeed, some years ago, managing security patches was only required on a small number of systems, mostly sensitive and very exposed ones (like Internet reachable servers). Applying patches as well as basic hardening rules were enough in order to get protected against so called "scripts kiddies" malicious activities.

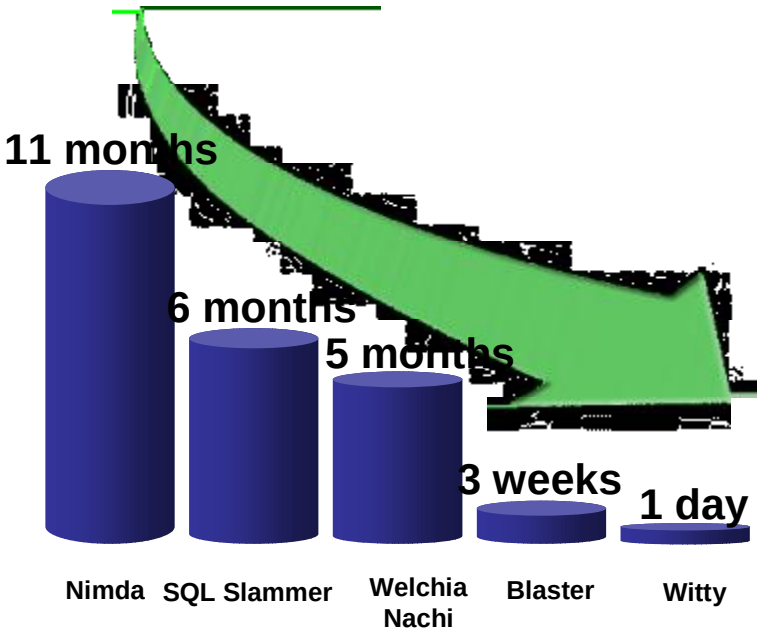
Unfortunately, things are no longer so simple today. The main reason is: viruses and auto propagating worms. The threats have indeed changed and are now automated with the apparition of worms that exploit security vulnerabilities inside software in order to hack unpatched systems, like CodeRed, Nimda or Blaster. Worse, the vulnerable perimeter is now really broader: today, it includes not only sensitive or exposed servers as before, but also every potentially vulnerable system (including users' PCs).

In this article, we will begin with an overview of some patch management tools in a Windows environment and then, we will present an example of a patch management policy applied in a particular big company.

Worms and viruses

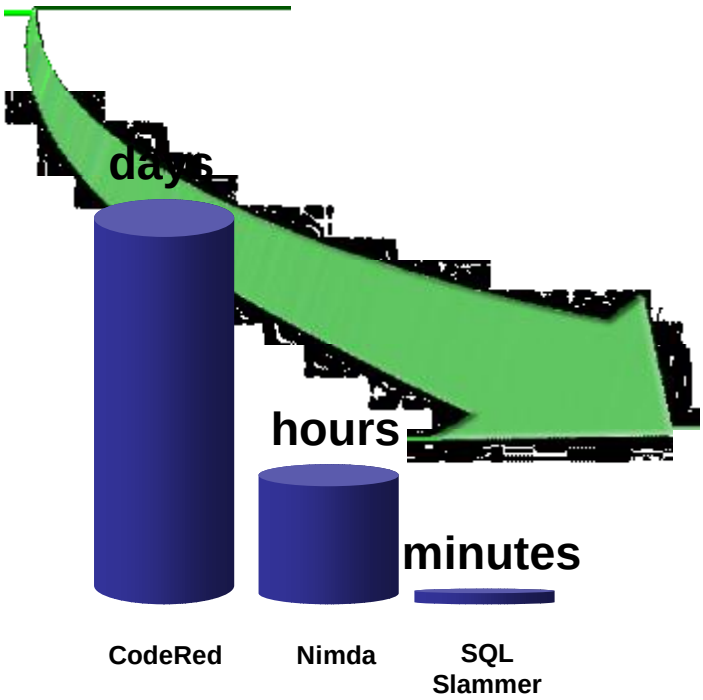
Until today, it was true that the impact on a single PC was low; however, when hundreds or even thousands of PCs are infected at the same time by worms scanning whole network ranges and looking for vulnerable systems, the entire internal network (LAN) of the company may be slowed down. Even worse, most viruses and worms have payloads that may be destructive (files deletion, etc.) or may send confidential information to other people (sending of files stored in Windows' "My Documents" folder or on the Desktop to all entries found in the user's address-book for example).

Moreover, a quick study of the worms having impacted many companies worldwide during the past years shows that the situation is becoming worse and worse:



Time elapsed between the release of a new vulnerability and its exploitation by a worm

In fact, for instance, from Nimda to Blaster, the gestation time of worms (time elapsed between when a new vulnerability is published by a software publisher and the apparition of a worm exploiting the vulnerability in question) is really shorter: 11 months for Nimda, 6 months for SQL Slammer and only three weeks for Blaster (we can also speak about Witty, affecting ISS software: one day only !).



Propagation speed of some worms

Moreover, the propagation speed (calculated from the apparition time of the first known instance of the worm in the wild and the time the highest number of contaminated systems is reached) is becoming higher and higher: from some days for CodeRed, to some minutes for SQL Slammer !

Clearly, it's nowadays urgent to do something if we don't want to spend most of our time (and money) reacting after a contamination. A solution is to define a patch management policy in accordance with the actual threats and with the business constraints of the company, and to implement it thanks to the tools that are available today.

Patch Management Tools

We saw before that security patches management is essential on a corporate scale. But the classical approaches, like the manual approach and the semi-automatic approach (example: Windows Update) are not enough. The patch management process needs to be automated to facilitate the vulnerability and patch watch, to be reactive to new threats, to test patches before deployment and finally to automate the roll-out.

A typical patch management policy

A typical patch management policy must address several questions: who ? what ? when ? where ? and how ?

Who: only administrators should be able to apply patches on computers. Users mustn't apply security updates on their workstations.

What: only updates from official vendors should be applied. A verification of the origin is mandatory. Moreover, the administrators have to test and approve the patches and reject useless or dangerous ones in the context of the company.
An automatic selection of necessary patches (among approved ones) is done, according to the OS version of the target computer: patches for Windows 2000 are not applied on a Windows XP computer, for example.

When: patches must be applied automatically, but with planning. For example, Microsoft recommendations are the following: critical patches should be applied within 24 hours, important patches within a month, moderate patches within 4 months and low-importance patches during the next 12 months…

Where: security updates must be applied on every computer. Even mobile laptops have to be patched, and, for that, they must return to the office periodically.
From where should patches be downloaded ? They should come from one or more central corporate server(s). Typically, the closest one is used.

How: patches are applied automatically, in the background, in order to be transparent to the users. If necessary, a unique reboot should be planned. Useful events should be logged during and at the end of the operation.

Such a patch policy management can be implemented using several patch management tools released by Microsoft. In the following paragraphs, we are going to analyze some of these tools.

MBSA

MBSA (Microsoft Baseline Security Analyzer) is one of the simplest tools released freely by Microsoft and aimed at system update level control. It is originally a Shavlik technology. This tool can be downloaded at the following URL:

http://www.microsoft.com/technet/security/tools/mbsahome.mspx

The latest version of MBSA is 1.2. A new version of MBSA (1.2.1) is needed for Windows XP SP2 compatibility: it offers deeper integration with SP2 security improvements.

MBSA 1.2.x can perform local or remote scans against Windows NT 4.0, Windows 2000, Windows XP, and Windows Server 2003. Along with Windows security patches, it supports also a great number of Microsoft products:

- IIS 4.0, 5.x, 6.0
- SQL Server 7.0, 2000
- IE 5.01+
- Exchange Server 5.5, 2000, 2003
- Windows Media Player 6.4+
- Microsoft Office (local scans only)
- MDAC 2.5, 2.6, 2.7, and 2.8
- Microsoft Virtual Machine
- BizTalk Server 2000, 2002, 2004
- Commerce Server 2000, 2002
- Content Management Server 2001, 2002
- Host Integration Server 2000, 2004, and SNA Server 4.0.

MBSA can be run in graphical mode (run `mbsa.exe`) or in command line mode (run `mbsacli.exe`). In this mode, you can use batch files to automate the tool. For example, the following script will scan a computer and log the results in an XML file:

```
set cname=%computername%
set uname=%username%
"C:\Program Files\MBSA\mbsacli.exe" /nvc /nosum /c %cname% /n
  IIS+OS+SQL+Password /o %cname%
copy "%userprofile%\SecurityScans\%cname%.xml"
  "\\%cname%\c$\Documents and Settings\%uname% \SecurityScans\"
```

MBSA replaces the old HFNetChk and MPSA tools. To emulate HFNetChk, you can launch MBSA with the following parameters:

```
mbsacli.exe -hf -?
```

How does MBSA work ?

Here are the steps followed by MBSA when you run it:

1 – MBSA analyses the security configuration of the scanned host. It detects the most common security configuration errors, like the following ones:

- FAT file systems
- Administrator accounts
- Weak passwords
- Dangerous running services
- File shares
- Audit policy
- ICF configuration (local scan only)
- Etc.

For a complete list of MBSA security checks, you can open the file `Checks.csv` with Excel in MBSA directory.

2 – Then MBSA downloads an XML security reference: it is in fact an XML file named **mssecure.xml**. MBSA can download this file from the Internet or from an internal MSUS server. From the Internet, MBSA tries to use these links:

http://go.microsoft.com/fwlink/?LinkId=18922
http://download.microsoft.com/download/xml/security/1.0/nt5/en-us/mssecure.cab (*version 3.32*)
http://www.microsoft.com/technet/security/search/mssecure.xml

The CAB contains a compressed version of mssecure.xml.
Note that you can also download the latest security reference from the following Shavlik links:

http://xml.shavlik.com/mssecure.cab (*version 4.0*)
http://xml.shavlik.com/mssecure.xml

Tip: If MBSA cannot download mssecure.xml, it uses the local copy (the latest downloaded version of the file). So you can download Shavlik's mssecure.xml file and use it for your MBSA scans ! But note that it is an unsupported tip…

3 – Next, MBSA analyses the patch level of the scanned host against the security reference.

4 – Finally, MBSA detects missing security patches and Service Packs and writes corresponding messages in its output.

The mssecure.xml file is a very interesting and somehow precious file: it contains the security updates list since 1998, along with their description data ! For each patch, the following information is provided:

- Description of the update
- Download path of the update
- Path, version and checksum of updated files
- Registry keys created or updated by the patch.

The mssecure.xml file contains also a history of previous hot fixes superceded by latest cumulative patches or Service Packs. And of course, the file is updated every time a new security update is released.

MBSA limitations

MBSA is a great and reliable tool, but it has some limitations: for example, when MBSA cannot confirm that a patch has been applied or not, it displays a "Note" message. This happens for example when products don't have detection criteria (MSXML for MS02-008 is in that case) or when there are more than one patch for a single product targeted at

a particular OS. It is in fact an mssecure.xml schema limitation. It is the case for example for DirectX 9.0 for Windows 2000 / XP / 2003 (MS03-030):

```
Note   MS03-030   Q819696
Please   refer   to   http://hfnetchk.shavlik.com/support   for   a
detailed explanation.  Refer to the section on Note Messages.
```

Sometimes, MBSA can only check Registry keys to determine if a patch is installed. For example, in MS03-037, common Registry keys are present for each vbe6.dll version, but file versions or checksums are different:

```
Patch NOT Installed      MS03-037        Q822150
File C:\Program Files\Common Files\Microsoft
Shared\VBA\VBA6\vbe6.dll has an invalid checksum and
its file version [6.4.99.69] is equal to what is
expected [6.4.99.69].
```

And when a non-security update overwrites files previously patched, MBSA reports the originally patched files as unsure:

```
Warning                  MS03-023        Q823559
File C:\Program Files\Common Files\Microsoft
Shared\TextConv\msconv97.dll has a file version
[2003.1100.5510.0] greater than what is expected
[2003.1100.5426.0].
```

Note that the language of the scanned computer determines also if checksum checks are performed (see /hf, /sum and /nosum options).

Scripting with MBSA

MBSA scans can be automated using scripts: you can perform large-scale scanning and enable low-rights end-users to check their own compliance without calling the helpdesk. For more information, see:

http://www.microsoft.com/technet/security/tools/mbsascript.mspx

You can download for example batchscan.js and rollup.js scripts, that scan an unlimited number of computers or IP addresses from an input file and compile the results into a single summary report (an XML file) that can be viewed with Internet Explorer.

Windows Update

Windows Update is an online checking and updating tool, that can be run in two modes: a manual mode, using Internet Explorer to browse http://windowsupdate.microsoft.com, and an automatic mode, using the “Automatic Updates“ service as the client part on Windows computers. Windows Update is ideal for patch management in small-sized companies.

When used in automatic mode, the "Automatic Updates" service, allowing an automatic and background update, requires the BITS service (Background Intelligent Transfer Service). This service utilizes the unused bandwidth to download the patches from Microsoft Website, in order to render the whole process totally transparent for the user.

Windows Update doesn't use the same mechanism than MBSA. The WU client verifies that every patch has been correctly installed on the local computer. To assume this, several checks are performed at different levels:

- Registry keys (located at `HKLM\SOFTWARE\Microsoft\Updates\Windows` *`[VERSION]`*`\SP`*`[X]`*`\KB`*`xxxxxx`*)
- File list on the disk
- Version and checksum of the files.

The update process runs in the background and is transparent for an ordinary user. Updates notifications are presented in a GUI to the current user only if he or she is a local Administrator:

This kind of notification looks exactly the same than with MSUS client (see below).

Microsoft Software Update Service

Microsoft Software Update Service (MSUS) is a free tool that can be downloaded at the following URL:

http://www.microsoft.com/windowsserversystem/sus/

Unlike MBSA, that scans several Microsoft products, MSUS manages only Windows security updates. It does not manage applications by now, but the next version of MSUS (version 2.0) should be able to do this.

The principle of MSUS is to have "Windows Update in your company": one or more internal servers host the security updates, then each time some new security updates are released, the administrator approves the necessary patches, and finally users automatically connect to one of the internal servers to download and apply approved updates.
Note that unlike WindowsUpdate, MSUS doesn't verify the serial numbers of the installed software… So don't be afraid to use it !

MSUS uses a Web-based administration tool (located for example at http://sus.your-intranet.com/SUSAdmin/), and requires IIS on internal servers. The IISLockdown and URLScanner tools are automatically installed on the IIS servers at setup time to secure them.

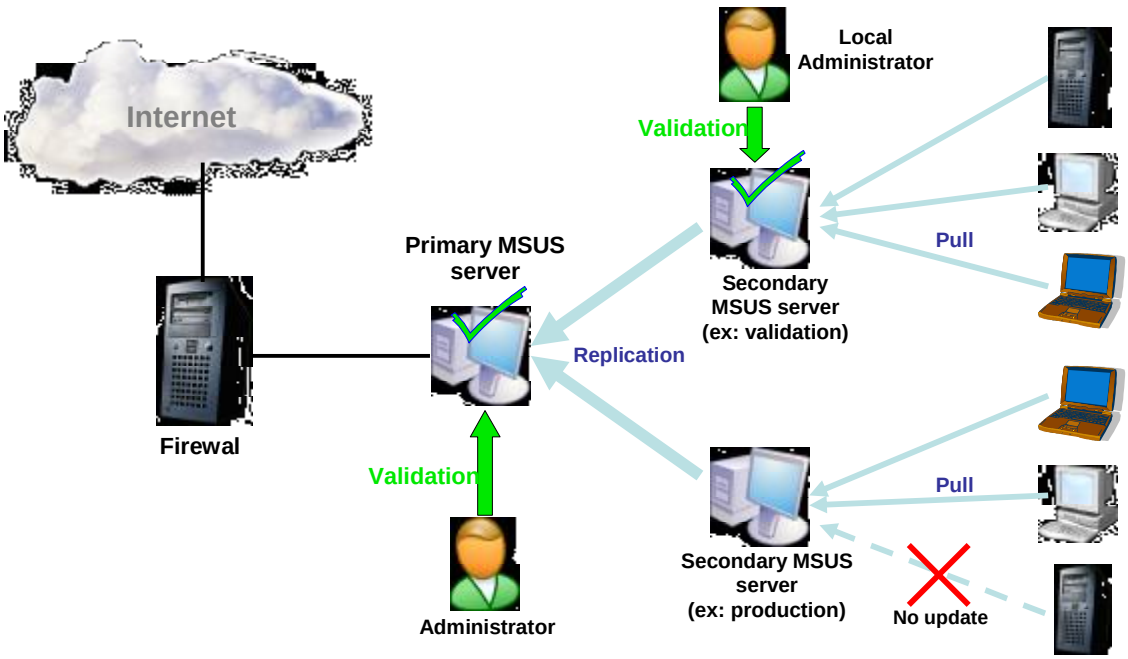
How does MSUS work ?

MSUS works a bit like MBSA, even if their internal protocol formats are different: MSUS requires also a security reference. Every day, at a specified hour, MSUS performs a synchronization process, following these steps:

1. MSUS downloads a security reference (XML files) located at:
   - http://www.msus.windowsupdate.com/msus/v1/aucatalog1.cab
   - http://www.msus.windowsupdate.com/msus/v1/aurtf1.cab
2. It validates the signatures of the CABs to verify that they actually come from Microsoft
3. It compares the reference with its local database content to determine which are the new updates
4. It downloads the new updates and verifies their signatures
5. It updates its synchronization and approbation logs
6. If the planned synchronization fails, SUS retries 3 times with a 30 min interval.

After this process, the administrators have to approve the new updates before they can be rolled-out and applied to the computers on the local network.

An advanced MSUS architecture looks like this:



You can see that you can configure several validation points, with different administrators validating different kinds of security updates, and several distribution points for different parts of your corporate network.

MSUS is a very powerful tool that can address the typical patch management policy we presented before, answering our "W*" questions:

Who: the « Automatic Updates » service runs with SYSTEM privileges, so it has sufficient privileges to apply patches in the background. If the user is an Administrator, he has the choice to apply the patches or not (see before). Ordinary users cannot install updates by themselves. Of course, once you have installed MSUS on your internal network, you should apply some filtering rules on your outgoing HTTP proxy server. The sites you have to authorize for the primary SUS server are the following:

http://www.msus.windowsupdate.com
http://download.windowsupdate.com
http://cdm.microsoft.com

And the sites that are forbidden for every other computer (servers and workstations), to avoid manual updates by users, are the following:

http://www.windowsupdate.com
http://windowsupdate.microsoft.com

What: MSUS can apply Windows security patches only, that is to say "security updates", Security Rollup Packages (SRP), Windows critical updates and Services Packs.
The SUS server checks the signatures of the updates to be sure they have been issued by Microsoft. Appropriate updates are then automatically selected (among the administrator-

approved ones), depending on the target computer (OS version, language, etc...). MSUS supports updates in 31 languages.

When: patches are applied automatically every day, at a planned time you can chose. A random delay between clients avoids simultaneous connections on the SUS server. The patches are also applied at boot time, if the specified time is exceeded.

Where: patches are applied on every computer that has an Automatic Updates client configured. An HTTP "pull" process is used to download the updates from the specified SUS internal server.

How: The updates are first tested and approved by the administrator. After this step, the updates are transferred between the primary SUS server to the secondary SUS servers, and from there, to every client, transparently, with bandwidth optimization, using the BITS service. Then the updates are automatically applied on every computer, in the background or not. Only if necessary (about 30% of the time), a unique reboot is performed. Then synchronization and approbation logs (in XML format) are written. We will examine these logs later in this paper.

According to Microsoft, each SUS internal distribution point can serve up to 15000 clients. You can also specify one or more logging points: you can distinguish each MSUS server between these two roles (distribution and logging point) in its configuration, though one server can have both roles.

The replication between internal SUS servers is also based on « Automatic Updates » and BITS services, allowing the use of the unused network bandwidth. But in spite of that, the transfer of updates can be very long (sometimes several days !). You have to consider that a whole set of updates for Windows 2000/XP/2003 represents about 600 Mo per managed language by now (Windows XP SP2, around 270 Mo, not included).

After approbation of patches by the administrator, the updates can have several status:

- New (no choice made by the administrator yet)
- Approved
- Not approved
- Updated (a new version of an update that has just been downloaded. It's a sort of "meta-patching", or patching of a patch)
- Temporary unavailable (unavailable associated update or unfound dependency)

Note that an update that has already been applied and that is then labeled "not approved" later by the administrator will not be uninstalled by the MSUS client. This won't be the case any more with WUS (Windows Update Services, the new name of MSUS version 2.0), that will be able to uninstall previously applied patches.

The MSUS client is already present in Windows 2000 SP3 or more, Windows XP SP1 and Windows Server 2003. For older versions of Windows, it can be downloaded as a stand alone setup from the following URL:

http://www.microsoft.com/windows2000/downloads/recommended/susclient/default.asp

It replaces the « Critical Update Notification » and note that it cannot be uninstalled !

The client and its parameters can be deployed on workstations by using GPOs or ADM files (WUAU.ADM, installed in %WINDIR%\INF). The client parameters are stored in the following Registry key:

HKLM\SOFTWARE\Policies\Microsoft \Windows\WindowsUpdate\AU.

These parameters are:

- The internal SUS server address (WUServer key);
- The update schedule (ScheduledInstallDay and ScheduledInstallTime keys);
- The download and installation types (AUOptions key): you can choose to notify for download and / or notify for install;
- Automatic reboot or not (NoAutoRebootWithLoggedOnUsers key). If the logged-on user is a local Administrator, he has the choice to reboot immediately or not. If he isn't an Administrator, the reboot is forced. You can also chose not to reboot automatically if there are any logged-on users and to wait for the next manual reboot.

Analysis of MSUS logs

MSUS includes two activity logs: a synchronization and an approbation log. Both of them are in XML format. Moreover, some client status logging is done server side: downloads and installations status are logged on the statistics server. As we've seen before, the statistics server can be different from the main SUS server, but if you have only one internal MSUS server, the statistics and the distribution server are the same computer.

Actually, MSUS uses IIS logs to write its own events. These logs are stored in the following files:

```
%WINDOWS%/system32/LogFiles/W3SVCx/exyymmddhh.log
```

Look for the string "`/wutrack.bin`" in these files to find lines written by MSUS. By analyzing the logs, we can obtain updates deployment statistics. Indeed, the status entries in IIS logs are in the following format:

```
/wutrack.bin?V=1&U=<Client_ID>&C=<client>&A=<activity>
&I=<item>&D=<device>&P=<platform>&L=<language>&S=<stat
us>&E=<error>&M=<message>&X=<proxy>
```

Example:

```
2004-08-16 16:09:55 127.0.0.1 GET /wutrack.bin
V=1&U=cebed56691e3194998b908b01ddbbf7c&C=au&A=w&I=ie60
x.internetexplorer6x.ver_platform_win32_nt.5.2.x86.en.
..3790...com_microsoft.q824145_ie_server2003.&D=&P=5.2
.ece.2.112.3.0&L=en-US&S=f&E=80190193&M=ctx%3D5&X=
```

```
040108110352351 80 - 123.123.123.123 Industry+Update+
Control 200 0 0
```

The “Client_ID” field is a unique ID affected to each computer on the network. The “Status” field indicates the status of the patch installation on the client. The possible values are:

- s: Succeeded
- r: Succeeded (reboot required)
- f: Failed
- c: Cancelled (by the user)
- d: Declined (by the user)
- n: No items (no update items were available for the client component)
- p: Pending

You can use a tool named “SUS Statistics Report tool” to produce reports from MSUS logs. This free tool can be downloaded at the following URL:

http://www.susserver.com/Software/SUSreporting/

Nevertheless, it is recommended to control the state of computers on the network after every update deployment by using an MBSA scan (see before), and to perform MBSA log analysis to detect computers on which a patch hasn’t been applied successfully.

Some client side logs are also generated during the update process. These logs can indicate if the patches installation was a success or a failure on the local client computer. The logs files are the following ones:

```
%programfiles%\WindowsUpdate\V4\IUHist.xml
%windir%\Windows Update.log
```

MSUS Tips and Tricks

Several useful tips and tricks could save you some time (and hair): for example, to force an immediate update detection of the AutoUpdate Client, follow these steps:

- Stop the "Automatic Updates" Service
- Check that the "AUState" registry value in the HKLM\Software\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update\ key is set to 2
- Delete the "LastWaitTimeout" registry value in the HKLM\Software\Microsoft\Windows\CurrentVersion\WindowsUpdate\Auto Update\ key
- Start the "Automatic Updates" Service.

You can also automatically approve new updates, without browsing to SUS Web administration interface and manually check some updates. Indeed, SUS stores its updates database in a dictionary object located in the following file:

```
C:\InetPub\wwwroot\autoupdate\dictionaries\ApprovedIte
ms.txt
```

Each update has a record that looks like:

```
com_microsoft.q311889_xp_5081,1@|0@|0@|2003-08-
01T15:06:35
```

The first "1" after the description stands for the status of the update: 0= unapproved, 1 = approved, 2 = new, etc… So a simple search and replace on this field updates MSUS database with new status information, bypassing the manual process. SUS Service will then read and update this file when it next does a synchronization with Microsoft Web site.

You can find more useful tools at the following URL:

http://www.susserver.com/Tools/

Other problems often occur on some networks: for example, the same updates are endlessly re-applied on a computer. In that case, detection criteria are incorrect, or update installation fails, or the installation succeeds but one detection element isn't recorded. To correct this, see related Microsoft Knowledge Base articles for identifying the detection criteria and make the modifications manually in the Registry or elsewhere.

Another problem occurs when some patches and the related cumulative patches are installed one by one on a computer. Unfortunately, MSUS 1.0 doesn't manage cumulative patches. You have to wait for MSUS 2.0 (WUS) and, when you approve a cumulative patch, un-approve individual patches embedded in it.

MSUS and SMS

MSUS can be integrated with SMS 2.0 and SMS 2003, thanks to the "SMS SUS Feature Pack". You can download it for free, and it includes the following tools:

- Security Update Inventory Tool
- Microsoft Office Inventory Tool for Updates
- Distribute Software Updates Wizard
- Web Report Add-ins for Software Updates
- Elevated-rights Deployment Tool.

With SMS SUS Feature Pack, you can deploy updates for other platforms and applications, not only Windows security patches. Moreover, you can track the status of the client installations: to do that, `mbsacli.exe` is pushed to each client and launched to perform a local scan (`mbsacli.exe /hf`), and finally the output is automatically parsed.

But SMS is expensive, and not every small company can afford it.

MSUS 2.0 and patch improvement

MSUS 2.0 will be called “Windows Update Services” (WUS). It should be released at the end of 2004. The main improvements will be:

- Download of approved updates only (currently, every update is downloaded on the MSUS server from Microsoft Web site before approbation)
- Management of cumulative patches
- Uninstallation of patches if they are un-approved
- Report generation
- Management of other Microsoft applications
- Unification of WindowsUpdate and OfficeUpdate into MicrosoftUpdate.

Microsoft also decided to release “non urgent” security patches each month (the second Tuesday of the month). This process will allow administrators to perform periodical test and roll-out cycles, and individual updates will then be deployed together.

Installation methods will be reduced from 8 (now) to only 2: MSI 3.0 and UPDATE.EXE.
The updates size will be reduced too. Today, the reduction is already by about 35%. With WUS, according to Microsoft, the reduction will reach 80%. Moreover, a “delta patching” technology, already working on WindowsUpdate, will be used, and the process will be improved with MSI 3.0.

The downtime will also be decreased. Today, the reduction of reboots is by 10% with Windows 2000 / XP / 2003. The reboots will be reduced by 30% with Windows 2003 SP1, and up to 70% with the next Server version, according to Microsoft.

After the study of patch management tools, we are going to see how a patch management policy can be implemented in the real world.

Example of policy applied in a company

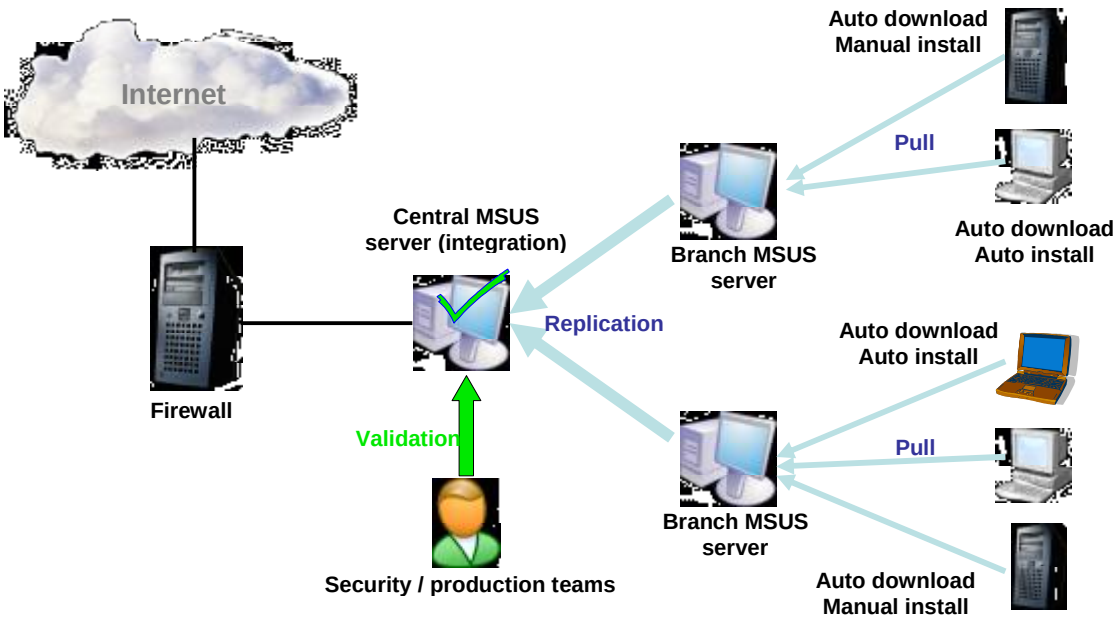
Accor Services is the second Accor Group’s largest international line of business. Their goal is to help companies and public institutions to improve performance by fostering the well-being of their employees and citizens. Present in 34 countries around the world, Accor Services is the worldwide leader in its line of business. As part of an important project of centralization of Accor Services’ Information System, a patch management policy, based on automated systems, has been defined and implemented.

This policy is defined as follows:

1. A new vulnerability, affecting one software used by the company, is published, associated (or not) with a patch release;

2. The central security team evaluates the risks, in terms of security, in the specific context of the company;
3. The patch is tested in a specific test lab, composed of IIS/ASP servers, Exchange and SQL Server servers, etc… VMWare is also used to validate some systems.
4. An internal advisory is sent to all the company's IT managers. It contains:
    a. Some details about the vulnerability;
    b. Information about the availability of exploits;
    c. Information about the availability of scanners allowing to identify vulnerable systems;
    d. Details on potential workarounds (if any);
    e. Information on known side effects when the patch is applied (based on the feedback gathered in public security mailing lists and during the tests performed previously).
5. The patch is then installed on a production-like test platform, that contains the common infrastructure systems installed in all deployed subsidiaries (i.e. Domain Controllers, Mail Servers, etc.);
6. After validation on the production-like platform, the patch is downloaded and installed in an automated way on all subsidiaries' PCs (except for some very specific exceptions);
7. Finally, patches are automatically downloaded on all servers, and then patches are applied as follows:
    a. The installation is launched remotely and by hand on common infrastructure servers;
    b. The installation is locally launched by hand for all other systems (done by each local IT team).

This policy also defines the maximum duration of each main tasks listed above, so that a critical patch could be deployed in less than 48 hours after validation (one working week for other patches).



On a technical point of view, the implementation has been done thanks to MSUS. A central server is installed at the Head Office and "slave" systems are deployed in each subsidiary. Thus, after the central validation of a patch, remote servers download validated patches and they are then installed on all systems, thanks to Windows GPO (Group Policy Objects).

Since its implementation with MSUS, early 2004, this policy has been improved many times. Our experience on this field allows us to provide the following advices:

- The scheduling of downloads by geographic areas is required for worldwide companies in order to lower the network activity;
- It's mandatory to always reboot the system after having applied a patch when a reboot is required;
- A specific policy has to be defined for mobile PCs (it may include a quarantine system of unpatched PCs or even an automated installation of the lacking patches).
- Regular checks of the patch level on all systems in the network (with MBSA in particular, or other tools) are important to assess the security level reached and detect non-corrected systems.

Conclusion

As explained above, no company can nowadays free itself from defining an efficient security patches management policy. Even if many tools are now available (we didn't detail in this article other tools like SMS), their installation first requires a serious study in order to reduce the vulnerability time of the company when a new security

problem is known, while not impacting the business of the company. Some tools exist for all kinds of company size: automatic Windows Update for small-sized companies, MSUS for medium-sized ones and MSUS or SMS + SUS Pack, for example, for big companies.

But hardening software is essential, especially when security patches are not applied yet. We can say that Microsoft seems to try to make its software more secure (more restrictive default configuration of Windows 2003, for instance) or less exposed (onboard personal firewall activated by default in Windows XP SP2, memory protection against buffer overflows in Windows 2003 and Windows XP SP2) as well as easing patch management (with the very waited MSUS 2.0).

However, new forms of "automated hacking" may appear in the future, making more than ever important to perform technical security watch in order to anticipate tomorrow's security problems.

References

- Patch management methodology
  - http://www.giac.org/practical/GSEC/Daniel_Voldal_GSEC.pdf
- Microsoft Patch Management
  - http://www.microsoft.com/technet/security/topics/patch/
- MSUS
  - Main Page
    - http://www.microsoft.com/windowsserversystem/sus/
  - MSUS Overview
    - http://go.microsoft.com/fwlink/?LinkId=6927
  - Useful tools and papers
    - http://www.susserver.com/Tools/
- Automatic Updates configuration
  - http://support.microsoft.com/default.aspx?kbid=327838
- Microsoft Strategic Technology Protection Program
  - http://www.microsoft.com/security/mstpp.asp
- Security Operations Guide for Windows 2000 Server
  - http://www.microsoft.com/technet/security/prodtech/windows2000serv/staysecure/