MICROSOFT WINDOWS SERVER UPDATE SERVICES(WSUS) REVIEW

Dominic White and Barry Irwin, Rhodes University

project@singe.rucus.net, B.Irwin@ru.ac.za

ABSTRACT

On November 16th 2004, Microsoft announced the availability of the Windows Update Service (WUS) Beta. This was half a year later than Microsoft's originally planned release date. This is hardly surprising given the growth in focus and increasing number of patch management solutions over the last year. With the window from vulnerability announcement to exploit release rapidly diminishing, patching has become one of the essential tools on the front-line of the security battle ground. Many organisations have been making do with Microsoft's Software Update Service (SUS) for the last year with many more still looking for a patch management solution. SUS was seen as a quick-fix due to its limited nature, leaving more than one administrator with handfuls of hair due to lack of reporting, limited product support and lack of flexibility. It is hoped WSUS can address these issues. This paper will detail the experience with WSUS from installation to use. It will start with an abridged description of WSUS's installation and configuration, and move on to a tour of its abilities. This is contrasted against SUS to better detail WSUS' improvements. The narrative then takes a turn for the technical, with the basic workings of WSUS explained, based on information gleaned from a live packet capture. The SOAP web service is dissected and found to have some security implications. The paper should provide an administrator with a glimpse of the inner workings of WSUS, and an overview of how it can fit into their organisation.

This work was undertaken in the Distributed Multimedia Centre of Excellence at Rhodes University, with financial support from Telkom SA, Business Connexion, Comverse, Verso Technologies, THRIP, and the National Research Foundation. The financial assistance from the DAAD foundation is hereby acknowledged.

KEY WORDS

Windows Server Update Services, Microsoft, WSUS, WUS, SUS, Patch Management

WINDOWS SERVER UPDATE SERVICES REVIEW

1 INTRODUCTION

On November 16th 2004, Microsoft announced the availability of the Windows Update Service (WUS) Beta. This date came half a year after Microsoft originally planned to release the service[1]. Then four months later on March 22nd 2005, Microsoft announced a new release candidate (RC) and a name change, WUS was to be called WSUS (Windows Server Update Services), and on June 8th 2005, Microsoft released WSUS to manufacturing. Given the growth in focus and solutions for patch management over the last year, the delays and changes are hardly surprising. With the window from vulnerability announcement to exploit release rapidly diminishing, patching has become one of the essential tools on the front-line of the security battle field. Many organisations have been making do with Microsoft's Software Update Service (SUS) for the last year with many more still looking for a patch management solution. SUS was seen as a quick-fix due to its limited nature, leaving more than one administrator with handfuls of hair. Microsoft has provided Systems Management Server (SMS) for enterprise patch management and other administration, however the price is not always appropriate, particularly for small to medium enterprises. It is hoped WSUS can address these issues.

This review will detail the experience with WSUS from installation to use. It will start with an abridged description of WSUS's installation and configuration and move onto a tour of its abilities. The narrative then takes a turn for the technical, with the basic workings of WSUS explained based on information gleaned from a live packet capture. The review is concluded with a list of useful WSUS resources.

2 WHAT'S NEW

WSUS introduces a number of new features over SUS, mostly due to the new back-end that Microsoft has implemented, WSUS uses the same back-end technology as Microsoft's Windows Update Service[2]. These new features are:

- Reporting Features. SUS provided no reporting at all, although the back-end implemented it. This lead to administrators having to rely on third party tools to derive what was going on in their organisation. WSUS provides a host of reporting features, partially fulfilling this much needed customer request.
- More Updates. In addition to operating system updates, WSUS now provides updates for Office XP and 2003, Exchange and SQL server. It still only provides support for Microsoft products and patches however.
- Update Filters. The administrative interface provides useful filter options to navigate the thousands of updates.
- Target Grouping. Machines can now be placed into different groups allowing different update approval options for each group. This is particularly useful for pushing patches to a test lab before large scale deployment.
- Improved Distribution. WSUS now allows express updates which use binary patching to push only those changes required (called deltas) rather than a whole file. It also supports an improved topology making distributing updates across the organisation easier. Finally, the download on demand feature can ensure that updates are only downloaded when necessary.
- Patch Options. WSUS now allows several new update approval options which enable WSUS to optionally check if a update is required, install the update or remove the update. WSUS also provides for automatic approval rules for specific updates and target groups.
- Improved Update Options. The new version of the BITS (Background Intelligent Transfer Service) includes new options which make installing updates on client machines less intrusive and disruptive.

- New Back-end. WSUS now sports a new back-end finished off with a SQL database. With the option of either SQL server or the free Microsoft Desktop Engine (MSDE).
- Secure Server Replication. Updates and configurations can be replicated between servers. In addition SSL connections can be used in server to server and server to client connections.

3 INSTALLATION

Installing WSUS is fairly straightforward. Microsoft have provided a good description of the process and its options in their WSUS deployment guide[3] along with a brief guide by WindowsSecurity.com[4]. Thus, this section will be fairly brief.

3.1 Topology

Before an installation an administrator should be aware of the topologies WSUS affords. With the new grouping options and the ability to distribute WSUS servers, several different topologies are possible. There are four basic models which can then be combined to form fairly complex systems if necessary. There are three primary components used: Microsoft Update (MU), the WSUS server and the WSUS clients, called Automatic Update(AU) clients.

3.1.1 Default

This is the 'normal' way of doing things, with the WSUS server receiving its updates and meta-data from Microsoft Update (in a process called synchronisation) and passing it on to its AU clients. The meta-data contains extended information about the update, such as the licensing and description, and can be separated from the actual update. See figure 1.

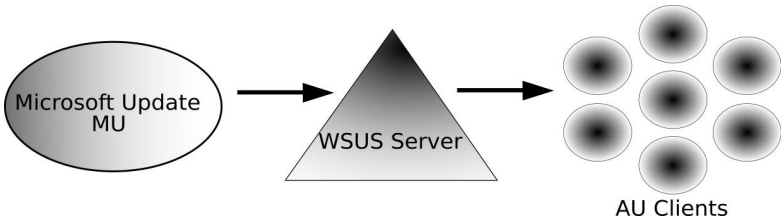

Figure 1: Default Topology

3.1.2 Grouping

WSUS's new grouping feature allows AU clients to be grouped separately. Each group can then have its own patch approval options. This is useful for testing, allowing patches to be pushed to a test lab before being pushed to the AU clients in a production group. A machine cannot be part of multiple groups, although this feature is often requested. The need for multiple groups is usually indicative of a poor topology and the additional complexities required to implement it is prohibitive. See figure 2.

3.1.3 Chaining

As with SUS, WSUS allows for a WSUS server to synchronise from another WSUS server rather than Microsoft Update. This is useful for creating a distributed hierarchical environment. Microsoft recommends that the hierarchy be no more than three levels deep, though they have tested it with up to five levels[3]. With this model a downstream WSUS server will inherit the approval and transfer setting of the upstream WSUS server. Such a topology can also be used as a disconnected architecture where WSUS's

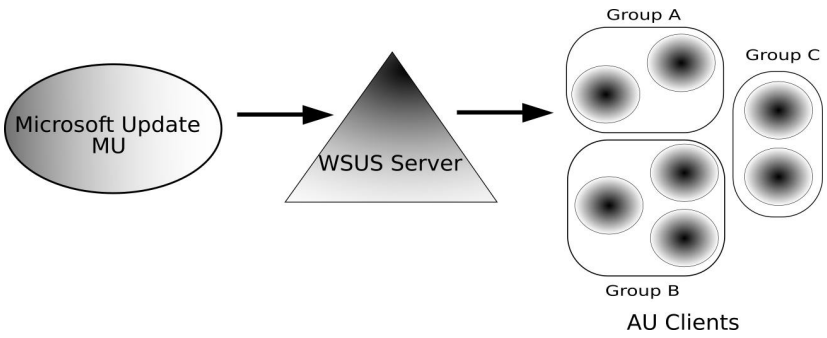

Figure 2: Grouped Topology

import/export update feature allows for updates to be hand-carried via sneaker net[1] from a connected WSUS server to a disconnected one.See figure 3.

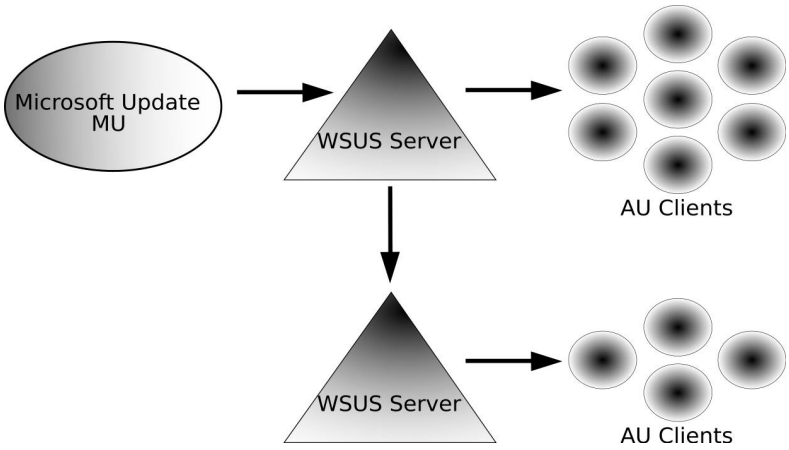

Figure 3: Chained Topology

3.1.4 Client Download

It is not always practical to download updates to the WSUS server for distribution. This is particularly true in mobile environments where the AU clients proximity to the WSUS server is unknown. In such situations the WSUS server can be configured to store only update meta-data. This allows the WSUS server to retain control over update approval without needing to store or distribute the updates themselves. The AU clients can then download the approved updates from Microsoft Update.See figure 4.

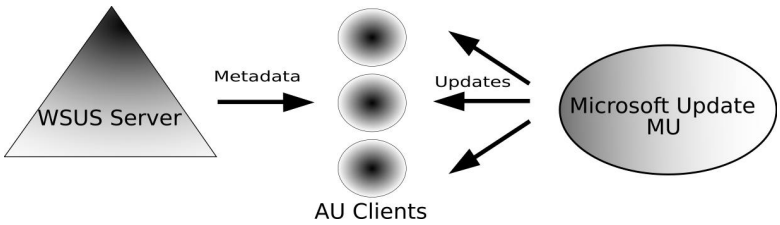

Figure 4: Client Download Topology

[1] Manually delivering patches to each machine without a network.

3.2 Requirements

WSUS requirements are fairly minimal and typical of the average server. The requirements are described in more detail in the WSUS Deployment Guide[3].

Microsoft recommends a 1GHz machine for less than 500 AU clients and a 2GHz machine for more than 500 AU clients. It should also have at least 1GB of RAM. WSUS requires either Windows Server 2003 or Windows 2000 Server with both requiring the .NET framework version 1.1 SP1[5, 6], BITS 2.0[7, 8] and IIS 6.0 and Windows 2000 Server requiring IE 6.0 SP 1[9]. Both should have 30GB of an NTFS file system free for updates and 2GB free for MSDE. Microsoft recommends using SQL Server over MSDE with more than 500 AU clients. MSDE for Windows Server 2003 (now named Windows SQL Server 2000 Desktop Engine or WMSDE) is distributed with the WSUS installer however Windows 2000 Server users will have to download it[10].

WSUS will only install to a Windows 2000 or 2003 Server with the .NET framework version 1.1 SP1[5, 6] and IIS 6.

Server installation is facilitated by a wizard, which will allow for an upstream WSUS server to be configured, instead of connecting to Microsoft Update. After a successful installation the WSUS administrative interface can be found at `http://server[:port]/WSUSAdmin/`, where [port] will only be used if WSUS was not installed to the default site, in which case the port will be 8530.

The automatic updates client has the same requirements as SUS and will only work on Windows 2000 with Service Pack 3 or later, Windows XP and Windows 2003. WSUS uses the automatic update (AU) client's self-update feature (present in SUS) to install the new AU client on each machine. However, this won't work on Windows XP machines without service packs installed, as it requires the SUS upgrade[11]. The client is first upgraded from the cab files found in \SELFUPDATE directory of the web server. Once upgraded it installs the new Windows Installer 3.1, BITS 2.0 and WinHTTP 5.1 which are needed to support the new configuration options WSUS affords. Windows XP SP2 already has an updated automatic updates client but will still self-update to the latest version. A more technical description of this process can be found later in this document.

4 CONFIGURATION

WSUS configuration is similar to SUS configuration. The behaviour of the WSUS server is controlled through the WSUS administrative interface while the behaviour of the AU clients is handled through group policy or the registry. This section provides a brief introduction to the various configuration settings available. Once again this is documented in greater detail in the WSUS deployment guide[3], with additional information available in the currently more complete SUS deploy guide[12].

4.1 Server

Server configuration is done via the WSUS administration page. Some options are shared with SUS and will not be covered in detail.

The WSUS server can be configured to synchronise with either Microsoft Update or another WSUS server, as discussed above. This requires information such as the server and proxy details and a schedule for how often the WSUS server should synchronise. The syntax for entering an upstream server is `http://servername[:port]`, with [:port] only used if the WSUS server is not using port 80.

WSUS now supports updates for Office, Exchange and SQL Server, as compared to SUS which had far fewer updates. Microsoft hopes to expand this to all of their products, and are looking into methods for securely distributing third party updates while maintaining the distribution security of signed updates[13]. Until then WSUS is not a complete patch management solution. This requires that the products, for which WSUS should distribute updates, be selected by adjusting the settings for which products, languages, and

class of update e.g. critical updates, security updates, service packs should be managed. Given the much increased number of updates there is an option to automatically approve certain classes of updates.

WSUS provides two methods for grouping computers. The first is server side targeting. This allows an administrator to manually place machines that have contacted the WSUS server into chosen groups. The second, more powerful, option allows the clients themselves to advertise to be put in a certain group. This setting is then controlled on the client either through group policy or registry settings. In both cases an administrator needs to create the group on the server.

The new distribution options afforded by WSUS allows for bandwidth consideration to be better accounted for. Deferred updates allow meta-data to be downloaded separately from the update files. This allows approvals to be disseminated and the update is only downloaded if required by an AU client connected to the WSUS sever (or a downstream WSUS server). The express installation is uses Microsoft's binary delta compression (BDC)[14] similar to FreeBSD's FREEBSD-UPDATE binary patching[15]. It allows for deltas to be sent to the AU clients. These deltas only contain information that should be changed within selected files rather than a replacement for the entire file. This can often yield large bandwidth savings by reducing patch sizes, particularly for security patches that usually only require a small change. Express installation does incur a cost in the form of a large initial download from to the WSUS server, as a delta for each possible version of the files needs to be distributed.

4.2 Client Side

The new background intelligent transfer service (BITS v2.0) and automatic update client allow for several new configuration options on the client side. The addition of these options appears to be Microsoft's response to the criticism of the limited options previous versions provided. In particular the fewer restarts and greater configurability should make the process more pleasant for the desktop user. These options can be modified in several ways; active directory group policy, local group policy or registry settings. These configuration methods are referred to as administrative policies, which are distinct from the user's configuration. A few of the options are common to SUS, therefore the focus will be on the changes and new options. Modifying these via group policy can be done by opening the group policy editor and navigating to *Computer Configuration/Administrative Templates/Windows Components/Windows Update*, after loading the windows update administrative template, WUAU.ADM (this will automatically be upgraded if done previously with SUS). Modifying the settings via the registry requires that the key *HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\Windows\WindowsUpdate[\AU]* be edited.

There are new options related to how update notification are displayed. These notification can occur either before downloading and installation, just before installation, or not at all. The first option prevents the automatic update client's user-interface from being locked when administrative policies are used to configure the client. This allows a local administrator to choose their own notification settings. The next allows for non-administrators to be included in the group of users allowed to receive update notifications.

With the introduction of grouping, client-side targeting is a method where an AU client will advertise which group it should be a member of, allowing clients to self-populate groups. Thus there is an option to specify which group the AU client should request membership of.

WSUS now takes more advantage of the agent on the AU clients and utilises a periodic check where an AU client will connect and allow the WSUS server to interrogate its patch status. This option is specified in hours. The AU client will connect between the specified time and a random negative 20% offset. Thus if the option is 10 hours the AU client will connect every 8 to 10 hours (as 10 hours - 20% is 8 hours) and this will vary randomly.

A separation has been made between updates that require a restart and those than don't. Non-restarting updates can be installed immediately without notifying the relevant user, if the user is configured to receive installation notifications. This allows the administrator to automatically install updates that don't require a restart without disturbing the desktop user unless an update requires a restart. This should reduce disruptions to the end user.

In the case of scheduled installations two new options have been provided. One allows a delay to be inserted before continuing with a scheduled restart, and the other allows the amount of time before the user is re-prompted for a scheduled restart to be specified. Minor, but occasionally useful changes.

The option to remove links and access to Windows Update was available in SUS, but is often overlooked and is therefore mentioned here. This will remove the link to Windows Update in the start menu and will prevent non-approved updates being installed from Windows Update. This setting can be found in the group policy editor at *Computer Configuration/Administrative Templates/Start Menu and Taskbar.*

In addition to administrative policies, the update client can be manipulated via the command line. This is done by running *wuauclt.exe* with command line switches. The two switches are: */resetauthorization* which will delete the client side cookie, which normally expires after an hour, and contains information such as the target group (more information on this can be found in section 7); and */detectnow* which will force the AU client to connect to the WSUS server and check for new approvals. When these switches are used together they must be used in the order they were mentioned i.e. *wuauclt.exe /resetauthorization /detectnow*. This is particularly useful for debugging machines and forcing an update.

5 PATCHING

The process of patching machines is done is six steps: synchronisation, approval, detection, distribution, installation and verification. This section will look at each step, and how WSUS supports it.

5.1 Synchronisation

During synchronisation meta-data is downloaded from a central distribution point, in this case Microsoft Update, and disseminated to other WSUS servers and AU clients. This process can also download the updates to the server allowing AU clients to fetch them locally, if WSUS has been configured to do so. WSUS uses BITS to transfer the meta-data and updates in the background and supports resuming the process if it is interrupted. The progress of a synchronisation is displayed on the front page of the WSUS administrative interface.

5.2 Approval

WSUS allows three types of approval to be applied to each update: detect only, install and remove. Currently no updates support the remove option, but will in the future as it is a function of the new Windows Installer. The update's approvals can apply to all machines or one group. A group can also inherit its approvals from the global configuration. The interface is far easier to use that SUS' allowing updates to be filtered by product, classification, approval, date received and by a text based search. The filtered updates can then be sorted by column.

5.3 Detection

Periodically an AU client will connect to the server and provide a list of platform details, installed updates, hardware and drivers. This is then used by the WSUS server to display which updates are needed by the AU client and which have been successfully installed. This is particularly useful for determining the patch status of an organisation. The frequency with which an AU client connects to the WSUS server is configured on the client (see section 4.2). This is where the AU client synchronises meta-data with its WSUS server.

5.4 Distribution

Updates are distributed over HTTP using the background intelligent transfer service (BITS), which supports resuming of interrupted downloads and dynamic throttling of downloads to use spare bandwidth. Updates

can either be downloaded from a local WSUS server or Microsoft Update depending on the topology (see section 3.1). Distribution has been made more flexible with the introduction of download on demand, where updates are only downloaded to the server when needed, and express updates, which make use of binary patching (see section 4.1).

5.5 Installation

Many of the patching improvements in WSUS are due to the new Windows Installer version 3. Microsoft has converged several patching methods in previous versions of Windows Installer into two installer-routines which are supported by the new installer[16]. This demonstrates a maturation of the format. The new MSI packages will also support uninstallation of updates[16], hence the new remove approval setting. In addition these packages will require less restarts and will support binary patching[16], hence the introduction of express updates. Other powerful switches have been added and more details can be found from Microsoft[17].

5.6 Verification

An important part of any patch management solution is the ability to verify that the patch was actually installed. In WSUS this is achieved through the same interface used for detection. The AU clients check in twice, once after downloading and installing updates and again after a machine restart in which the update installation is completed. During this check in the AU client return information on the status and progress of the update's installation, which after the restart, should be completed.

6 REPORTING

The single largest problem with SUS was its complete lack of reporting. WSUS offers four reports officially labelled as such. Namely:

1. Status of Updates
2. Status of Computers
3. Synchronisation Results
4. Settings Summary

The two most useful are a breakdown of updates or computers which allows an administrator to drill down to see statistics for groups and individual AU clients or updates.. These reports can be filtered by approval and groups, and can then be sorted by each column. This is not the only reporting in WSUS as many other screens provide reporting features, such as the computer and update screens. On the back-end all of the information is stored in a SQL database, allowing ad-hoc queries to be address through third party tools (such as Microsoft Systems Management Console). This is a great improvement over SUS and should satisfy most users, however the inflexibility of the reporting may frustrate some administrators seeking to generate specific reports.

7 PACKET CAPTURE

To get a better look at how WSUS does its work, Ethereal[2] was used to perform a packet capture of the communications between an AU client and the WSUS server. This revealed several improvements over SUS. Further, it demonstrated the working of WSUS which have not been published in much detail as yet. No study was done of the interaction between an upstream WSUS server and a downstream one, but this may be done as future work. The testing here was performed on a variety of WUS and WSUS pre-releases and so some of the bugs mentioned may have been resolved.

[2] http://www.ethereal.com/

7.1 Steps Performed

The relevant tasks performed during the packet capture were:

1. A new Windows XP SP1a AU client is joined to the active directory domain.
2. AU client self-updated.
3. The new AU client installs the Windows Installer and BITS updates, required a restart.
4. Logged in with some automatic update activity. The logged in administrator was not informed, although the icon appeared briefly. A restart was required.
5. Logged in and 24 new updates were downloaded.
6. Updates were installed, restart was required.
7. WSUS server synchronised with Microsoft Update.
8. *wuauclt.exe /detectnow* was run from command line on the AU client.
9. One critical update detected, downloaded and installed.
10. The same critical update was detected, downloaded and installed multiple times until approval was revoked on WSUS server.

7.2 Resulting Network Traffic

By comparing the resulting packet capture to the steps performed above, the interactions between the WSUS server and the AU client was discovered. Below is a chronological list of recorded HTTP request traffic between the WSUS server and AU client, and its analysis.

Step 1 (from the above section 7.1)

- /IUIDENT.CAB - This stands for 'Industry Update Identification' and is how the client's version is identified. This .cab file along with the rest below was time-stamped by Verisign and signed by Microsoft. If this were the first communication of a machine with a WSUS AU client rather than a SUS AU client (e.g. Windows XP with SP2) then these first three steps are not seen and the traffic would start with a call to WUIDENT.CAB.

Step 2

- Once it is determined that this is a SUS client, the self-update from step *2* is performed. The client is instructed to download the relevant .cab files (starting with WACOMP.CAB which contain version information for the individual client files) of the new automatic update client. In this client configuration the files were stored in /SELFUPDATE/AU/X86/XP/EN/ on the IIS server.
- /WUTRACK.BIN - After the self update the client requests WUTRACK.BIN with a parametrised query string. With SUS, the request of WUTRACK.BIN was used for reporting and statistics on the patch process. The parameters provide information on aspects of the clients behaviour, including platform, activity and the KB of the patch being installed (more information can be found on page 83 of the SUS deploy guide[12]). This method is how third party SUS reporting tools were developed (e.g. K. Hoover's[18] or D. White's [19]). With WSUS the item and activity parameters are not used but platform information is provided. This was the only request to WUTRACK.BIN seen in the whole capture and appears to be left for backwards compatibility.
- /WUIDENT.CAB - This stands for Windows Update Identification and contains AU client version information. This request includes a datestamp as a parameter.
- /WUSETUP.CAB - This contains an .inf and .cat file which contain setup information, such as dll version and registration information, for the new automatic update client. This request also includes a datestamp as a parameter.

- From here the new automatic update client communicated with the WSUS sever using a SOAP based web service. The format used to describe the SOAP method calls is: `[returned information] MethodName (passed information)`

  - `[config] GetConfig`
  - `[auth cookie] GetAuthCookie`
  - `[cookie] GetCookie (encrypted(auth cookie))`
    After this the returned cookie is encrypted and sent as the preamble to all future transactions. This cookie will contain information such as the target group of the AU client and expires after an hour.
  - `RegisterComputer (a SOAP XML file is passed with the full platform information)`
    The full platform information consisted of a description of installed hardware, the operating system version and build and some information on additional software. Some of this information may be considered sensitive.
  - `[required update ID's] SyncUpdates (system information, e.g. platform information, installed updates and drivers)` ...

    This is how the WSUS server knows what updates are needed on the client. This method is be called several times. The first time it is called the client sends empty update ID parameters. The last time it is called it contains strings of hardware drivers installed on the client, which coupled with the platform information in this call and the previous gives an almost complete picture of both the hardware and software the machine is running.
  - `[metadata] GetExtendedUpdateInfo (update meta-data)`

    This includes information such as the EULA and description of each update.
  - `[confirmation] ReportEventBatch (meta-data and sync updates status)`

    Information about the status of the client registration is returned. The client passes a large XML file to the server here detailing the status of the updates and once again providing platform information.

Step 3

- The first batch of updates is then downloaded as per point 3 above. In this case it is the Windows Installer 3.1 and BITS 2.0 updates. Once installed these will allow the full WSUS functionality to be used. Files are downloaded from subdirectories of the /CONTENT/ virtual directory in chunks, presumably to allow resuming of downloads if the process is interrupted.

  - `[confirmation] ReportEventBatch (update download status)`

    Information about the status of the download of the patches. This is sent before the update is installed, but after it is downloaded. According to the WSUS deploy guide, the AU client should request meta-data from the WSUS server again after downloading but before installation[3]. This is to ensure that approvals revoked during the download are not ignored. However there were no separate requests representing this, but it is presumed it would occur here.
  - `[confirmation] ReportEventBatch (update installation status)`

    Before the client restarts and after the updates have been installed another report is made. A separate report is made for each installed update.

Step 4

- After a restart the behaviour seen in point 4 is seen. No notification was received by the logged on administrator, this conflicted with how group policy had been configured. It was assumed that updates for immediate install were being installed as that option *had* been activated (see section 4.2). However a restart was required, which should not happen if this activity was as a result of immediate updates, as they do not require a restart. This could be due to a misconfigured update, the AU client ignoring group policy settings or an undetected misconfiguration, but it could not be determined at the time. It resulted in the following use of the web service:

  - `[confirmation] ReportEventBatch (update installation status)`

    This is a report on the, now complete, installation of the updates installed before the reboot.

  - `[location of update file] GetFileLocations (update ID's and file digests)`

  - The updates are then downloaded. Once again the AU client should check that none of the approvals for the downloaded updates have been revoked during the download. It is presumed that this check would be part of the GetFileLocations method.

Step 5

- The machine is then restarted and 24 updates are available, after which these calls are made:

  - `[confirmation] ReportEventBatch (update installation progress)` ...

    This is presumed to be reporting on the status of the installation of updates from the previous point.

  - `[update location] GetFileLocations (update ID's and file digests)`

  - The updates are then downloaded and installed.

  - `[confirmation] ReportEventBatch (update installation progress)`

Step 6 and 7

- The machine is then restarted during which time the WSUS server synchronised with Microsoft Update. Another call to ReportEventBatch is made to report the successful installation of the updates.

Step 8

- Running *wuauclt /detectnow* resulted in:

  - /WUIDENT.CAB
    /WUSETUP.CAB

    These request are made with a datestamp as a parameter.

  - `[required update ID's] SyncUpdates (system information, e.g. platform information, installed updates and drivers)` ...

  - `[metadata] GetExtendedUpdateInfo (update meta-data)`

  - One update is then downloaded and installed.

  - `[confirmation] ReportEventBatch (update installation progress)`

These are the items of interest. Partial packet captures are available from this site `http://singe.rucus.net/masters/files/WSUS-packetcapture.tar.gz` for further analysis, and a large more complete capture is available on request.

7.3 Analysis

From the information above, a pattern of behaviour can be mapped. This pattern was inferred from the behaviour of three different machines performing several updates.

When the AU client first contacts the WSUS server it makes two requests, each with a datestamp as a parameter. The files are returned time-stamped and signed.

1. wuident.cab
2. wusetup.cab

After this all future interactions (apart from BITS downloading the updates) are done via a web service.

After which if the AU client does not have a cookie or its cookie has expired the following handshake is made with the WSUS server:

1. GetConfig
2. GetAuthCookie
3. GetCookie
4. RegisterComputer

If the AU client still has a valid cookie, the above does not occur. The cookie is then prepended to all future transactions.

If the WSUS server has synchronised with an upstream server since the AU client's last synchronisation, a new synchronisation is performed. This looks like:

1. SyncUpdates
2. GetExtendedUpdateInfo
3. ReportEventBatch
4. Updates are downloaded.

If the AU client does not need to synchronise but has pending updates, a call is made to:

1. GetFileLocations
2. Updates are downloaded.

A reporting call is made after every action, and would be made after an update sync, update download and update installation. After the updates are downloaded the call is made:

1. ReportEventBatch

After the installation of the updates another report is made:

1. ReportEventBatch

If a restart is required to install any of the updates, another call is made after the machine has rebooted and, presumably, installed the updates.

1. ReportEventBatch

7.4 Packet Capture Summary

7.4.1 Interface

On the whole WSUS seems to be better designed. It utilises an open SOAP based web service, keeps track of each interaction and provides far more information on the patching process. SUS on the other hand, required third party log analysers to interpret an obscure query string. The use of a standard web service should make it easier for third party extensions to be created. The large amount of information generated should allow for many different reporting options beyond what WSUS currently offers.

7.4.2 Security

There are two security worries here, the first is disclosure of sensitive information and the second is interference with the patch process. The downside of the extra information mentioned above is that a lot of information about client machines is being sent as clear-text, this information includes a list of hardware, installed drivers and some software being used. There is enough information to allow an attacker to build a replica system to test attacks on. This is a worry, but it can be mitigated by good network design. The second worry is less troublesome as a man in the middle attack (which the cookie exchange may be vulnerable to) would not be able to circumvent the security of the signed patches.

8 RESOURCES

There are several fairly useful resources for WSUS available. Several were quite useful while writing this document.

1. Microsoft's WSUS page[20]
2. The WSUS Wiki[21]
3. SUS Server[22]
4. Patch Management Mailing List[23]

9 CONCLUSION

WSUS is definitely a large step in the right direction. It has many great improvements over SUS, which seem to indicate that Microsoft is listening to the consumer and responding to their community security needs. The interface is easy to use and provides some great functionality. The extra features provided on the client-side are equally welcome. Microsoft has developed a good architecture from which their patching strategy can be better managed and built upon. The use of an open SOAP web service allows for extensibility and thought has been put into keeping patch distribution secure. There are some problems with the amount of plain text system information that WSUS makes available on the network which may provide valuable insight to an attacker. However, the most notable problem is that WSUS still only supports a limited range of Microsoft's products and is sorely lacking support for third party updates. Given the number of vulnerabilities in popular products such as Winamp, Adobe Acrobat and Winzip, WSUS does not provide a complete patch management solution. Some of these problems are resolved in Microsoft's Systems Management Server (SMS) and others in third part products available from third party vendors.

References

[1] M. J. Foley. (2004, July) Microsoft delays by a year delivery of two new patching systems. Microsoft-Watch. [Online]. Available: http://www.microsoft-watch.com/article2/0,1995,1656785,00.asp

[2] (2005) Microsoft windows update. Microsoft. [Online]. Available: http://www.windowsupdate.com/

[3] "Windows update services deployment white paper," White Paper, Microsoft, Tech. Rep., November 2004. [Online]. Available: http://www.microsoft.com/windowsserversystem/wus/deployment.mspx

[4] A. Zinman. (2004, November) Windows update services review. WindowsSecurity.com. [Online]. Available: http://www.windowsecurity.com/articles/Windows-Update-Services-Review.h%tml

[5] (2004, March) Microsoft .net framework version 1.1 redistributable package. Microsoft. [Online]. Available: http://go.microsoft.com/fwlink/?LinkId=9104

[6] (2004, August) Microsoft .net framework 1.1 service pack 1 for windows server 2003. Microsoft. [Online]. Available: http://go.microsoft.com/fwlink/?LinkId=35326

[7] (2004, November) Microsoft windows update services bits 2.0 beta for windows 2000 server. Microsoft. [Online]. Available: http://www.microsoft.com/windowsserversystem/wus/betaeulaWin2k.mspx

[8] (2004, November) Microsoft windows update services bits 2.0 beta for windows server 2003. Microsoft. [Online]. Available: http://www.microsoft.com/windowsserversystem/wus/betaeulaWin2003.mspx

[9] (2002, September) Download internet explorer 6 service pack 1. Microsoft. [Online]. Available: http://go.microsoft.com/fwlink/?LinkId=22355

[10] (2004, December) Microsoft sql server 2000 desktop engine (msde 2000) release a. Microsoft. [Online]. Available: http://go.microsoft.com/fwlink/?LinkId=35713

[11] (2002, June) Automatic updates june 2002. Microsoft. [Online]. Available: http://go.microsoft.com/fwlink/?LinkId=22338

[12] "Software update services deployment white paper," White Paper, Microsoft, Tech. Rep. [Online]. Available: http://www.microsoft.com/windowsserversystem/sus/deployment.mspx

[13] M. Semilof. (2004, March) Microsoft taking steps to integrate wus with windows. TechTarget. [Online]. Available: http://searchwin2000.techtarget.com/qna/0,289202,sid1_gci956193,00.html

[14] Microsoft, "Binary delta compression," Tech. Rep., mar 2004. [Online]. Available: http://www.microsoft.com/downloads/details.aspx?FamilyID=4789196c-d60a-%497c-ae89-101a3754bad6

[15] C. Percival, "An automated binary security update system for freebsd," Master's thesis, Computing Lab, Oxford University, Oxford, 2003. [Online]. Available: http://www.daemonology.net/freebsd-update/binup.html

[16] P. Thurrott. (2004, April) What you need to know about windows update services. WindowsITPro. [Online]. Available: http://www.windowsitpro.com/Windows/Article/ArticleID/41969/41969.html

[17] Microsoft. (2005, March) Description of the new features in the package installer for windows software updates. kb 832475. Microsoft.

[18] K. Hoover. (2004, June) Ken's sus scripts. [Online]. Available: http://pantheon.yale.edu/~kjh27/sus-scripts.html

[19] D. White. (2004, December) Sus reporting tools. [Online]. Available: http://singe.rucus.net/sus/

[20] (2005) Microsoft windows update service home. Microsoft Inc. [Online]. Available: http://www.microsoft.com/wsus/

[21] M. A. Khaleel. (2005) Windows server update services wiki. [Online]. Available: http://wsus.editme.com/

[22] S. Korman. (2005) Susserver.com. [Online]. Available: http://www.susserver.com/

[23] J. Chan. (2005) Patch management mailing list. Shavlik Technologies, LLC. [Online]. Available: http://www.patchhmanagement.org/