Global Information Assurance Certification Paper

Using and Evaluating Windows Software Update Service
GSEC 1.4b

Abstract:

This paper describes the installation and use of Microsoft's Software Update Service (SUS) for the deployment of Operating System patches. It will feature an in-depth discussion SUS' features, installation, configuration (both client and server side), and built-in security. Additionally, it will provide an analysis of SUS, its potential affect on an environment and any shortcomings found during its evaluation.

By John Ives

Table of Contents

Introduction

The need for patches

When the MS SQL Slammer worm hit on Saturday Jan 25th 2003, it exploited a bug that had been patched seven months before with Microsoft Security Bulletin MS02-039 (June 24th 2002)[1]. Similarly, when Code Red wreaked havoc on IIS servers in the summer of 2001, it exploited a known, and previously patched bug. Examples like this highlight the need for proactive patching, a need that often goes unmet in many small to medium corporations and Academic environments where the expense of patch management has often been an impediment to security. Unlike the home environment where Microsoft's Automatic Windows Update Service doesn't generally pose a problem, the need for a managed environment has caused many administrators to turn off the Windows Update Service in the workplace.[2] With this in mind, Microsoft released Software Update Services[3] (SUS) a free version of the Windows Update website that can be run on a local IIS server and which allows an Administrator to approve patches before they are automatically deployed to users. This paper will look at the design, use and installation of SUS. Additionally, there will be an analysis of its ease of use, how it aids in computer management and its security implications.

Frequently hackers prey upon systems that have been poorly patched and are poorly managed, knowing full well that any compromise is unlikely to be quickly discovered and seriously pursued. With this in mind, Microsoft's Software Update Service (SUS), has the potential for being of substantial aid to IT support staff by helping automate the distribution of approved patches and by maintaining logs of which computers have and have not been patched. Additionally, since it is a free add-on to Windows 2000 Servers its initial support costs can be relatively minor, though its ongoing support costs must also be evaluated when considering the deployment of any such tools.

What is Software Update Service?

SUS is a locally run and managed version of Microsoft's Windows Update site (http://windowsupdate.microsoft.com). It runs on IIS 5.0 or above and allows clients running Windows 2000 (Professional, Server or Advanced Server) with Service Pack 2 or higher with the Windows automatic updating software to automatically query the SUS server using HTTP to determine if there are any patches needed by the local computer. Once the server has been checked and the client computer has determined that there are new patches to install, it can download them either from the local SUS server or from one of Microsoft's Windows Update Servers and install them. The downloading of the patch uses the Background Intelligent Transfer Service (BITS) which uses idle network bandwidth in order to prevent any network disruption. Once the patches have been downloaded, the installation can happen either automatically (immediately

[1] "Port 1434 MS-SQL Worm"
[2] Smith
[3] http://www.microsoft.com/windows2000/windowsupdate/sus/default.asp

or at a scheduled time) or the next time an administrator logs into the local computer.

A key part of SUS' usefulness is that the result of any patch download and installation is logged to a statistics server, also using HTTP. This gives administrators and support staff the ability to keep track of an individual computer/server's patch status and evaluate if systems are not being patched correctly. In loosely managed and small to medium sized environments, where the use of a more robust program like Microsoft's System Management Server[4] (SMS), which provides software deployment and asset management tools, is too expensive to manage, SUS provides a potential solution.

Key Features

For overwhelmed IT staff's in moderately sized environments, SUS has several key features that can help lower the maintenance costs of running Windows 2000 and XP. The first and clearly most important feature is the ability to roll out tested patches automatically without having to visit every machine personally. Second, SUS and its client component allow the installation of the patches to occur at scheduled times to minimize user disruption. Third, SUS has built-in logging that allows administrators to find out if any computers failed to get patched. From a security perspective, the final key feature is that all update packages are digitally signed. By checking this digital signature the SUS server and Automatic Update clients are able to detect if the files have been tampered with or corrupted, ensure that the files installed are the ones distributed by Microsoft.[5]

Evaluation Environment

To accompany the writing of this paper, a test environment was also established to allow for a more accurate evaluation of the software and its potential uses and shortcomings (results of this evaluation will be found in the section titled "SUS Analysis"). Because of its relevance to the evaluation of SUS, the settings employed in the test environment will be outlined in this paper. With only a limited number of computers available for testing, SUS was configured in a virtual environment using VMware Workstation 3.2.[6] Though virtual machines are just that, virtual, they do a good job of allowing people to closely approximate real computers. For this evaluation one of the important features of using VMware was the ability to set-up the SUS server with its own IP address so that it could participate in a real network with real clients. Because of SUS' required memory footprint (512MB) and the limits of VMware workstation, specifically that it will only allow the use of 1GB of RAM for all virtual machines, being able to bind the virtual machine to a real network made it possible to test more workstations.

[4] http://www.microsoft.com/smserver/default.asp
[5] "Software Update Services Overview," 4
[6] http://www.vmware.com/products/desktop/ws_features.html

Installation

Server Requirements

From a hardware perspective, the requirements to run SUS and automatic updates are very reasonable at both the client and the server sides. SUS requires a Windows 2000 (with Service Pack 2) or higher server running IIS. The hardware required to run SUS is a 700MHz Pentium III with 512MB of RAM and 6 Gigabytes of available disk space.[7] The client computer simply needs to be running Windows 2000 Professional, Server or Advanced Server with Service Pack two or higher or any version of Windows XP and Windows Server 2003.

Required IIS Components

Though the default installation of IIS will contain everything necessary for SUS, the Deployment Guide specifies that the only essential components for SUS to function are Common Files, the Internet Information Services Snap-In and the World Wide Web Server.[8] During this evaluation, IIS was installed with only these base components.

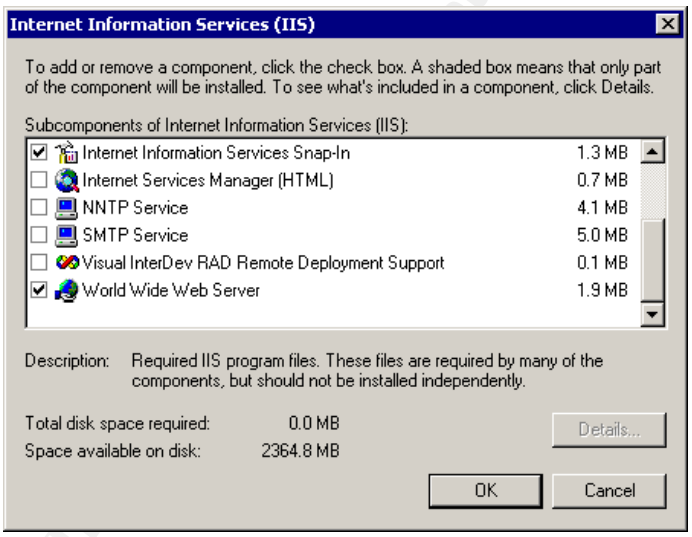

Image 1: Minimum IIS components

Restricting IIS components to the bare minimum components is a generally accepted security measure and, between the Deployment Guides assurance and the evaluation system running in this configuration, SUS displayed no problems in this configuration.

SUS Website

Prior to installation it is necessary to plan how SUS is to be installed and incorporated into any existing IIS environment. Because of the IIS Lockdown and Urlscan policies that are installed with SUS (and which will be discussed in the section on SUS Security), Microsoft recommends installing SUS on a

[7] "Deploying Microsoft Software Update Services," 7
[8] "Deploying Microsoft Software Update Services," 67

dedicated server. In order to follow this recommendation, stop all websites in the Internet Information Services Snap-in and allow SUS to install itself to a new web site bound to port 80.[9] One benefit of doing it this way is that the new website won't have any of the default files and folders associated with IIS, thus eliminating some of the files targeted so often by scripts and worms such as Nimda which attempted to exploit vulnerabilities through the directories /scripts, /_vti_bin, /_mem_bin and /msadc.[10]

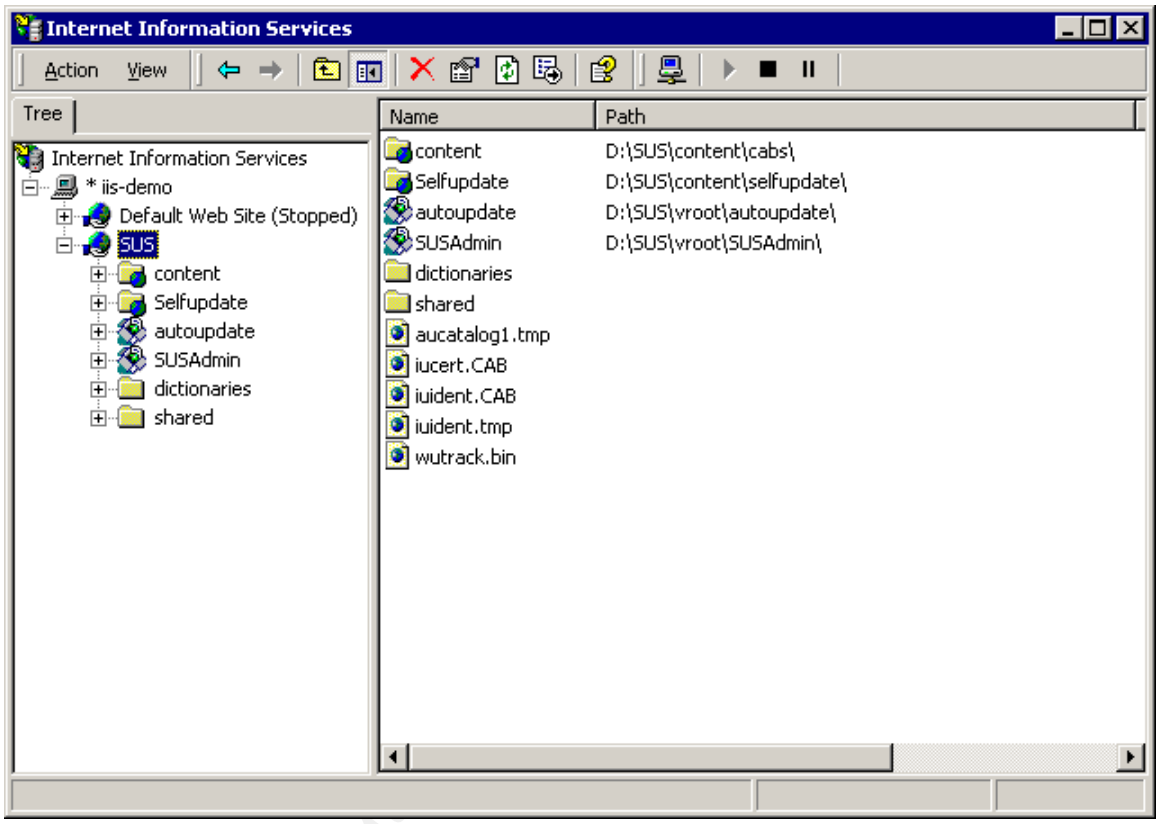

Image 2: SUS installed as its own Site

If, however, it is decided to install SUS to an existing site SUS will be installed in the site bound to port 80. For the sake of the accompanying evaluation, SUS was installed as its own website, by stopping the default site prior to installation.

Installing SUS is very simple requiring only a few minutes after the setup file has been downloaded. Running the SUS installation program (currently SUS10SP1.exe) on the server begins the installation process using an MSI (Microsoft Installer) package. After agreeing to the End User License, administrators are given the option of choosing a typical or custom installation. To ensure that the SUS was installed to a (virtual) drive set aside for that purpose a custom installation was selected, where this not the case SUS would automatically install in the disk with the most free space. [11]

[9] "Deploying Microsoft Software Update Services," 67
[10] "CERT® Advisory CA-2001-26 Nimda Worm"
[11] "Deploying Microsoft Software Update Services," 72

The first choice to make during a custom installation is where the website is stored on the server's disk drives. This is a relatively benign decision; however, it is important to remember that the disk SUS is placed on will need to have space for all of the required patches downloaded onto the server. The amount of space required is, in large part, affected by the other option on that same screen, whether the updates would be stored locally or would be obtained from Microsoft's Windows Update Server. This is an important option because it affects internet bandwidth usage and the SUS server's storage requirements. Additionally, the decision can be affected by a company's proxy server configuration. Because Automatic Update clients cannot get to updates through a proxy server requiring authentication while the SUS server can,[12] if a company is using this sort of proxy server they will need to choose the option to store the files locally. Because, it reduces bandwidth usage and because it is less likely to create issues with any potential proxy servers that may be installed in the future, the updates were maintained locally during the accompanying test and would certainly be deployed in this same manner in a production environment.

After the location, it is possible to determine which languages would be supported. Depending upon the environment, most companies, particularly the small to medium sized ones that will find SUS interesting will be able to limit this selection to one language, however it is possible to choose any number of supported languages. On the evaluation machine, with only English selected the total storage required for this installation, with current updates as of March 19th, 2003 was 268MB for the website and its content, however Microsoft claims that a complete installation with every language would be approximately 600MB.[13]

From an ongoing management perspective, how updated patches are to be handled is perhaps the most important decision made during the SUS installation process. Periodically, Microsoft updates previously released patches. When this happens, SUS can be configured to either automatically approve the patches or require the administrator to approve it a second time. From an ease of management perspective, it is easier to have SUS automatically approve the patch, but there is no guarantee that the updated patch won't introduce a new problem in a given environment. Conversely, because most updated patches fix problems, the testing and approval of a previously approved patch may not be worth the time required. Ultimately, this setting will vary from deployment-to-deployment.

Once the updates setting has been made, SUS is able to complete its installation, after which it becomes possible to modify any of the installation choices made and configure any proxy settings by using the SUSAdmin page (http://<server>/SUSAdmin).

[12] "Deploying Microsoft Software Update Services," 16
[13] "Deploying Microsoft Software Update Services," 15

Configuration

Because no product is completely 'install and forget,' it's important to configure both the server and the client appropriately. Without entering the appropriate settings an SUS server will not automatically synchronize and clients would be unable to update automatically.

Server

Configuring SUS is a straightforward endeavor that is accomplished through the use of a web page on the SUS server. This interface looks a lot like the Windows Update Server website and is the primary interface for all SUS configuration and management. These options include the languages supported, the file storage location, and proxy settings. Most of the configuration options were already covered as part of the custom installation; however the options to set the computer's name and configure proxy settings for Internet access are only available from the "Set Options" screen.

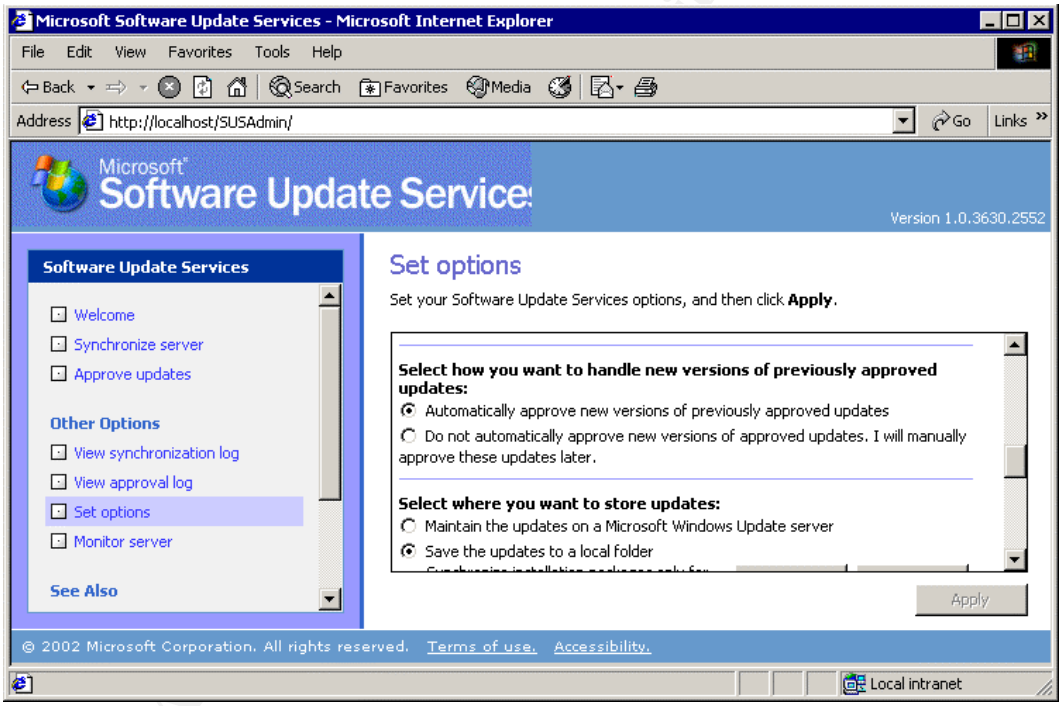

Image 3: SUS Options Setup

One of the most important options that can be set for the server is the "Synchronize Server" screen found on the left hand side of the SUSAdmin page. From this page it is possible to either "Synchronize Now" or set a "Synchronization Schedule." As the name implies, "Synchronize Now" is a manual synchronization with the Windows Update servers which immediately downloads the current version of Aucatalog1.cab.[14] In contrast to the manual synchronization, "Synchronization Schedule" opens a dialog box in a separate

[14] "Deploying Microsoft Software Update Services," 14

window (see Image 4) allowing the Administrator to schedule regular synchronizations based upon time and day of week (or everyday if so desired).

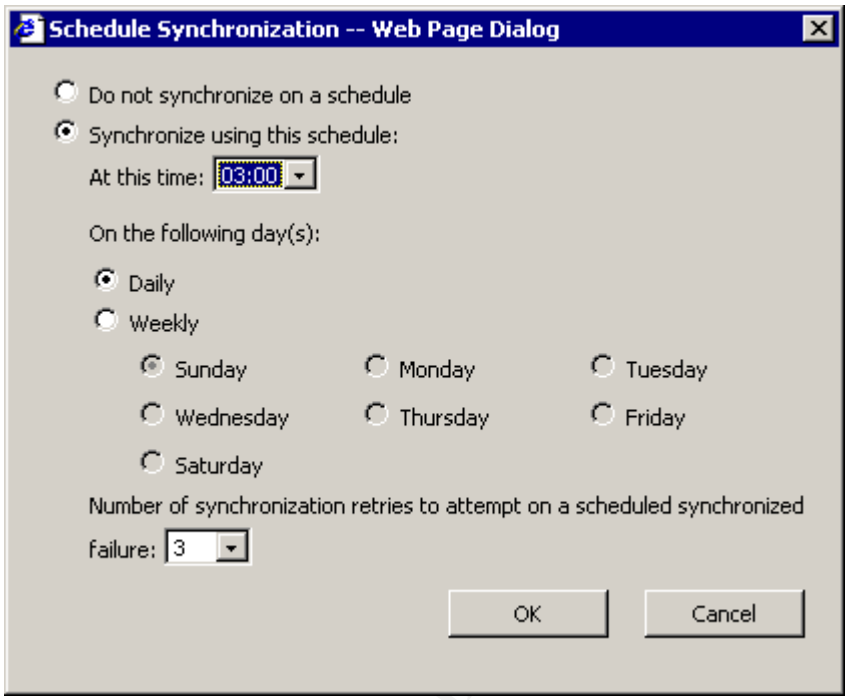

Image 4: SUS Synchronization Schedule

Client

For the client computers it is necessary to configure the update schedule, location and whether updates are automatically installed or just downloaded. This configuration can be accomplished through edits to the registry or Group Policy (either domain level GPO's or local Group Policy). In order to change a large number of computers quickly using Active Directory Group Policy is the best method. However, when dealing with systems in a non-Active Directory environment Local Group Policy or registry files (.reg – which automate registry changes usually done by hand) can also be used effectively.[15] When deciding between Local Group Policy and registry files it should be noted that Local Group Policy changes have less of a chance of causing problems in the registry while reg files are faster. For the sake of this paper, Local Group Policy will be discussed; however those wishing to use one of the other methods should see Microsoft Knowledgebase article Q328010, "How to Configure Automatic Updates by Using Group Policy or Registry Settings."

Using Local Group Policy there are two policies that can be configured as part of Automatic Updates (see Image 5), "Configure Automatic Updates" and "Specify intranet Microsoft update service location." Within the "Configure Automatic Updates" policy are options for the Automatic Update behavior which includes

[15] For more information on using registry files see Microsoft Knowledge Base Article 310516, "HOW TO: Distribute Registry Changes to Computers in Windows XP" (Note that while this specifies XP, I have successfully tested this same approach on Windows 2000 computers.)

"Notify for download and notify for install," "Auto download and notify for install" and "Auto download and schedule the install." To truly automate the installation of patches, it is necessary to set the computer to "Auto download and schedule the install." Also in the "Configure Automatic Updates" policy are the options for scheduling the updates. In terms of scheduling, the updates can be scheduled for once a week or for everyday, for these tests, everyday was selected. In terms of time, the Automatic Update can be configured for any whole hour (relative to the client).

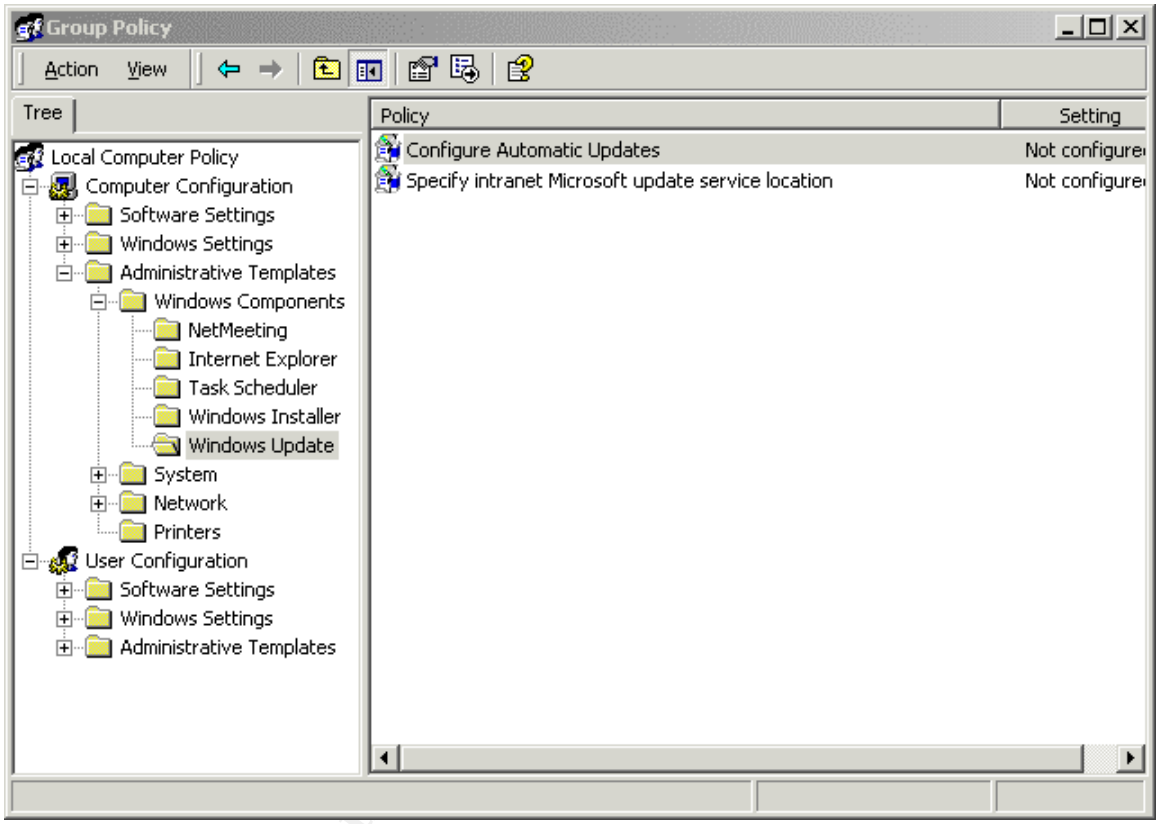

Image 5: Configuring the Automatic Update Client

When configuring the policy "Specify intranet Microsoft update service location," there are two options to configure, "Set the intranet update service for detecting updates" and "Set the intranet statistics server." Both of these fields take a HTTP URL that can be the same machine. This is significant because in a wide scale deployment with multiple update servers, statistics about patch state could be consolidated to one server for simplified management.

SUS Security

Over the years, Microsoft has been criticized for releasing products which installed with little or no security by default. However, with SUS Microsoft has started to take some of that criticism to heart installing IIS Lockdown in order to secure the IIS server and using digital signatures to verify the validity of any patches at multiple stages of the update process. These security measures,

though not perfect, go a long way toward protecting the underlying IIS server and the clients it serves.

IIS Lockdown Tool with URLScan

If IIS Lockdown and URLScan have not already been installed on the IIS server, they are installed and configured as part of the SUS installation.[16] IIS lockdown is a template driven tool available from Microsoft that is used to turn off unused IIS features and services.[17] ISAPI extensions are programs that IIS use to process files with particular extensions such as .asp. By default IIS includes programs called ISAPI Extensions that handle files like Active Server Pages (asp & asa), Server Side Includes (shtml, shtm & stm), Internet Printing (printer), Remote Password Changes (htr) and access to the Indexing Service (ida & idq).[18] In SUS, IIS Lockdown removes all but the ASP ISAPI extension, meaning that attempts to exploit any of the other ISAPI extensions will be unsuccessful.

URLScan is a filter which screens all server requests and filters them to ensure only valid ones are processed.[19] During the installation of SUS, URLScan is installed using the urlscan_dynamic.ini rules. An analysis of the urlscan_dynamic.ini rules file reveals that it only allows the HTTP verbs GET, HEAD and POST, thereby restricting the ways in which the client can interact with the server to the essential methods. Additionally, urlscan_dynamic.ini is configured to reject any web request accessing some executables (.bat, .cmd and .com), the Index Server, remote password changes, Server Side Includes, internet printing, some legacy databases access tools (.idc) and various static files (.ini, .log, .pol, and .dat). This provides a fallback position to the IIS Lockdown tool such that if one of the items secured with that tool should become re-enabled, it will still be rejected by URLScan.

Digital Signatures

In order to ensure the integrity of all patches distributed by SUS they are digitally signed by Microsoft. When SUS downloads a file from Microsoft (or another SUS server) it verifies the signature on the file and, if the certificate on the files doesn't match Microsoft's, the file is immediately deleted. Likewise, the client performs the same check of every file it downloads from the SUS server, and again deletes the file if it fails the verification. By checking the files at each transfer, the file is verified to be undamaged and free from malicious alteration by someone either injecting a new file it into the transfer as part of a man-in-the middle attack or modifying the SUS' \Content\cabs files.

Patch Management

Once SUS has been installed and configured, it becomes possible to test (because it is outside the scope of this paper a brief discussion of some of the

[16] "Deploying Microsoft Software Update Services," 10
[17] "IIS Lockdown Tool"
[18] Fossen, 156-157
[19] "Urlscan Security Tool"

common parts of patch testing will be discussed in Appendix B), approve and deploy patches. Though the testing is handled differently within every organization, the synchronization, approving and deployment are all handled through the SUSAdmin pages.

Approving patches

After the patches have been synchronized and any new patches are downloaded, it is then possible to approve patches from the SUSAdmin page. To approve patches, click on the "Approve updates" link in the left frame, which will bring up the "Approve Updates" page. On this page every available update is listed as either "New," "Approved," "Not Approved" (if it has been listed previously but hadn't been approved) or "Updated." To approve a patch, simply click on the check box next to the patches name and click on the "Approve" button (see Image 6).

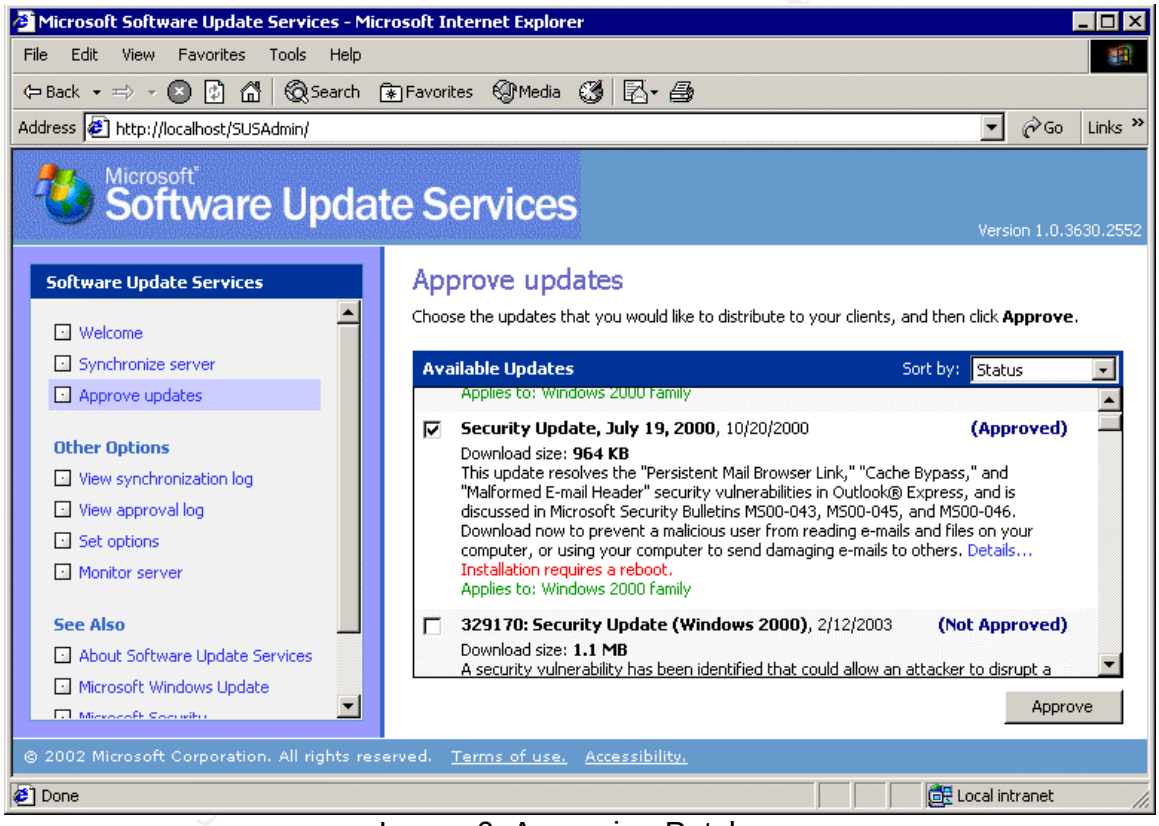

Image 6: Approving Patches

Once a patch has been approved it is possible to un-approve it, by un-checking the check box and again clicking on the "Approve" button.

Distributing patches

Using SUS the majority of the work involved in the distribution of patches is handled by the Automatic Update client. As discussed previously, the Automatic Update Client can download patches and install them automatically. Since patches are installed on a set schedule, there always is the possibility that either someone is working on their computer when the computer is set to install

patches or that the computer is off. To cope with this Microsoft has built in features to handle this possibility.

There are several potential scenarios related to users on end machines when the patch installs and Microsoft's strategy for dealing with them is outlined in its "Deploying Microsoft Software Update Service" document. Ultimately, the computer's behavior depends upon the client Automatic Update Client configuration as well as who is logged into the computer at the time. For instance, if the Automatic Update client has been configured to "Download the updates and notify me when they are ready to install," the computer will not install them until a system administrator has logged in and clicks on the Automatic Updates icon in the system tray to install the software.

If on the other hand the administrator would like more automation in the patch deployment process, they would configure Automatic Update to "Automatically download the updates, and install them on the schedule that I specify" and schedule a time. In this scenario the computer would automatically download the patches and notify the first administrator that logged in that there are patches to be installed. If the Administrator doesn't proceed with the installation at that time, the computer will wait until its scheduled installation time. When the scheduled time comes around, if an administrator is logged in they are notified of the impending installation and given the option to stop it (see Image 7) which will cause it to be delayed until the next scheduled time, otherwise the installation proceeds.

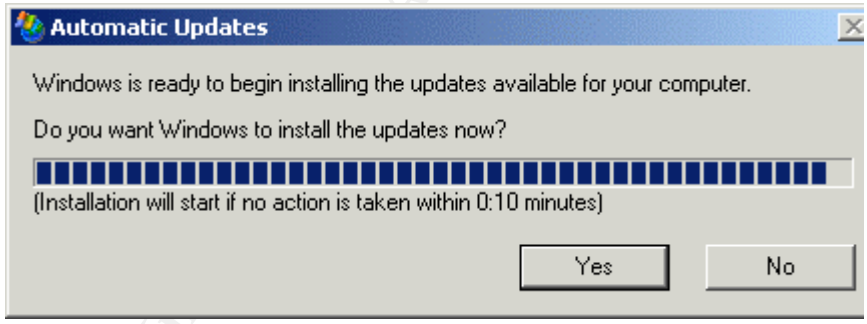

Image 7: Impending hotfix installation for Administrators

Once the patches are installed the system will automatically restart if required by the patch unless someone is logged in at the time. If there are logged in users, they are prompted with a message that the system will be restarting in 5 minutes and they are given the option to do an immediate restart. If the user is an administrator, the also have the option of stopping the restart process all together with the "No" button.

In the event that the computer was off at the scheduled installation time it will, by default, wait until the next scheduled time. However, if the administrator has set the RescheduleWaitTime value to 1 (minute) (either in the registry or using group policy), the computer will start the patch installation 1 minute after the Automatic

Update service starts. Since this should happen before anyone has logged in, the computer will not need to notify anyone and will restart immediately.[20]

Logging

As mentioned earlier, one of the more promising features of SUS is the ability to better manage workstations and servers in environments without many resources. In order to actually manage the patches on machines, it is necessary to assure that machines are actually being patched as desired with SUS. By design, there are two primary methods of handling patch auditing in SUS, the SUS/IIS logs and computer event logs (specifically the System Event log). While neither of these logging options provides all of the answers, they compliment each other with each filling a specific need. The Automatic Update client has the ability to log events to a "Statistics Server" using HTTP to create entries in the IIS logs with the relevant information. These statistical logs make it easy to get a fairly quick overview of which systems are being patched by making it possible to centralize the certain patch related information to one central location. Besides the Statistical logging, each computer maintains a System Event Log which (as the name says) tracks system events including patch installation and restarts. The System Event log is a key part of any troubleshooting because it includes information that wouldn't be included in any statistical logs (like an inability to contact the SUS server), but which are relevant.

IIS/SUS Logs

Between the two types of logs associated with the Automatic Update process, the IIS/SUS server logs provide the fastest access to general information about if a fix has been downloaded and installed. These logs are kept in \%SystemRoot%\System32\LogFiles\W3SVC1\ by default. Since this is simultaneously the SUS and IIS logs, the log entries are interspersed with any web traffic connections and are based upon the end machine's IP address, though using one of the fields called a Ping ID it is possible to track it even in an environment where IP addresses are dynamically allocated. Because, the IIS logs are key to evaluating and using SUS a more thorough discussion of using them will be found in the Log Analysis section later in this paper.

Event Logs

During both SUS' synchronization process as well as the clients patch installation and restart process, several success and failure events are written to the System event logs which can be used for auditing and troubleshooting. Because these types of events are, without the use of third-party software, kept on each individual computer, they are not a particularly effective way of tracking the overall state of systems patches across a large number of computers. Despite the difficulty in managing the event logs, they do act as a secondary logging mechanism to the SUS statistics server and the also provide detailed information about any client or server issues. With this in mind, they are ideally suited to the

[20] "Deploying Microsoft Software Update Services," 55-56

troubleshooting process, because they include events denoting failures to connect to the SUS or synchronization servers as well as any problems with any digital signatures. Additional information about the Client and SUS server event logs can be found in the "Deploying Microsoft Software Update Services" whitepaper.

SUS Analysis

For the purposes of this paper, SUS was evaluated on its ease of administration (both from a deployment and a logging perspective), the upkeep it requires, its effect upon an environment's security, its deployment costs and any shortcomings or problems that were encountered either in the SUS software itself or its documentation.

Remote Patch Administration

As discussed earlier, SUS is managed through a web interface. Access to that web page (http://<server>/SUSAdmin) requires administrator access,[21] however, during testing it was noted that, if basic authentication to the SUSadmin page was used, it occurred without the use of SSL/HTTPS. While this may, in rare circumstances, be acceptable in a limited testing environment, production SUS servers should require HTTPS for access to the administrator pages. Basic Authentication, as spelled out in RFC 2617, "…is not a secure method of user authentication…" which "…results in the essentially cleartext transmission of the user's password over the physical network."[22] Since it is necessary for the user accessing the SUSadmin page to have administrator level rights, the result of someone logging in to manage the SUS server is the cleartext transmission of an administrator password across the network which could then be intercepted by attackers.

In order to require SSL it is first necessary to install a server certificate which can either be purchased from a certificate authority like Verisign[23] or created locally using Microsoft's Certificate Services. Administrators wishing to obtain and install a server certificate for the SUSAdmin pages should read Microsoft Knowledge Base Articles 228821 "Generating a Certificate Request File Using the Certificate Wizard in IIS 5.0" and 228836 "Installing a New Certificate with Certificate Wizard for Use in SSL/TLS." Once a certificate has been installed the SUSAdmin page can be secured by right click on the SUSAdmin folder and choose "Properties." From the properties window, select the "Directory Security" tab and clicking on the "Edit" button in the "Secure communications" section and checking the "Require Secure Channel" box.

Besides using SSL, some environments may also wish to use IP address restrictions to help control access to the SUSAdmin pages. While IP restrictions are weaker than SSL and they cannot prevent an attacker from using the administrator's IP address while the administrator's computer is off-line (either

[21] "Deploying Microsoft Software Update Services," 10
[22] Franks, 18-19
[23] http://www.verisign.com

from a normal shutdown or through an attack), they do make any attempt to access the SUSAdmin pages more difficult. To apply IP restrictions to the SUSAdmin pages do this use the “Edit” button in the “IP address and domain name restrictions” section of the same properties tab used to require SSL. From this window the default restriction should be changed to ‘deny access’ and IP address can be entered for each of the locations charged with handling patch approval. In doing this it is important to remember that the local shortcut to “Microsoft Software Update Services” created in the “Administrative Tools” is a link to the SUSAdmin website so localhost should be included as an exception.

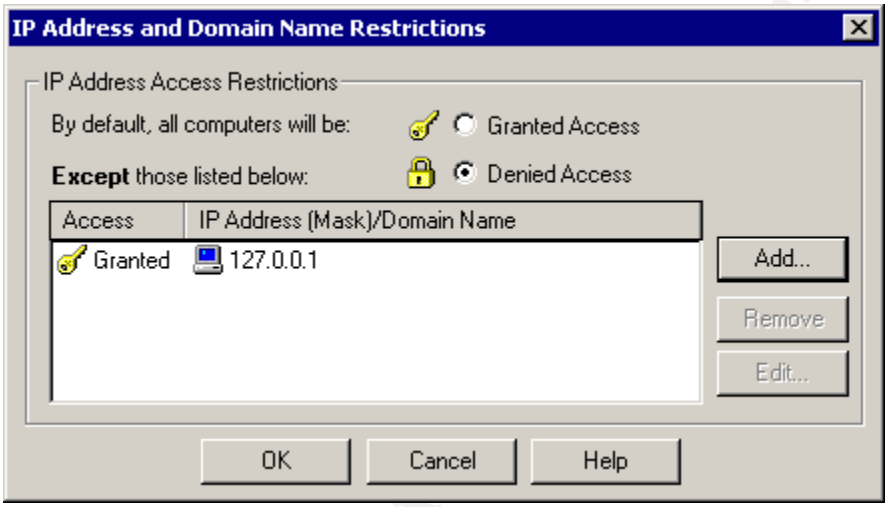

Image 8: IP Address Restrictions for SUSAdmin

Log Analysis

Potentially, system administrators and/or support staff can use the SUS logs to find out the patch status of a machine remotely without the use of any special software, unfortunately the reality of using these logs is that they are very cryptic to the naked eye making manual analysis difficult and there are currently few tools to help automate the process.

Manual Analysis

Below is a single (wrapped) line from the IIS log that demonstrates the complexity of the log entries used for this sort of statistical analysis. Additionally, analyzing this line will give a feel for the type of information that can be retrieved from the statistics server.

> 2003-02-20 00:41:26 192.168.254.130 - 192.168.254.128 80 GET /wutrack.bin V=1&U=51bd360ea8a8cb4e860c878b903147d6&C=au&A=w&I=win2k.windows2000.ver_platform_win32_nt.5.0.x86.en...2195...com_microsoft.q328310_w2k_5815.1_10_101_0&D=&P=5.0.893.2.0.1.0&L=en-US&S=s&E=00000000&M=&X=030220004534997 200 Industry+Update+Control

Looking at the beginning of this line, “2003-02-20 00:41:26 192.168.254.130 - 192.168.254.128 80 GET /wutrack.bin” provides the following information: date (2003-02-20); time (00:41:26 GMT); client IP address(192.168.254.130); client side username (here that is reported as a simple “-“ since there was no login

occurring); server IP address (192.168.254.128); server TCP port used (default HTTP port 80); the HTTP verb used (Get); and the file that is used specifically for this tracking (wutrack.bin). The last two entries on this line "200 Industry+Update+Control" are similarly straight forward with 200 meaning that the access to wutrack.bin was completed successfully and Industry+Update+Control being the client used for the connection (i.e., the Automatic Updates client). Between these two sections lies the bulk of the information that is needed to understand what happened and to track the update process for this client.

Using Microsoft's "Deploying Microsoft Software Update Services" guide it is possible to interpret the remainder of the above log entry (for the most part). Administrators should refer to the deployment guide for a complete description of all the possible parameters and messages that can be returned by the client. A breakdown of the above example line is as follows:

Though most of the entry is identifiable using the deployment guide the very first entry "V" is missing from the guide. A search of microsoft.public.softwareupdatesvcs led to a message from Don Cottam, of Microsoft, saying that the V=1 means the client computer is "running a version of iuctl.dll and iuengine.dll that were released after SUS1.0 was released," probably the result of updating the controls at Microsoft's Windows Update website.

"U" is a unique client identifier called a Ping ID that enables someone to identify the number of unique computers getting updates from the server. In this case the Ping ID is 51bd360ea8a8cb4e860c878b903147d6.

"C" is the Client and represents "the client type issuing the status response."[24] The "au" found in the Client entry signifies an action like downloading or installing patches.

"A" is the Activity with the value "w" denoting a download.

"I" stands for Item and identifies what the Client is working with. Here that item is "win2k.windows2000.ver_platform_win32_nt.5.0.x86.en...2195...com_microsoft.q328310_w2k_5815.1_10_101_0." Reading the text of the item string indicates that item being downloaded concerns Knowledgebase article Q328310 and that the item is specifically for Windows 2000.

"D" signifies the Device. There is little in the documentation for Device, but it does say that for non-device items the value will be empty as it is here.

"P" stands for Platform and includes information such as the OS version, build, product suite available on the server (e.g. Advanced Server, Terminal Services, etc.), whether it is a domain controller, server or client, and the processor architecture. In this case 5.0.893.2.0.1.0 can be

[24] "Deploying Microsoft Software Update Services," 83

interpreted (in brief) as Windows 2000 Professional, build 893, running on a 32bit Processor.

"L" is the entry for Language and is obviously US English.

"S" is the Status and in this case (S=s) it was successful.

"E" signifies Error and is an 8-digit hexadecimal value that in this case contains 00000000 because it isn't being used.

"M" is a Message field that can include information about any errors and is empty when not used.

"X" is the Proxy field and provides a timestamp of the status report in the format of "YYMMDDHHMMSSmmm"

As stated earlier a more complete handling of the event log requires a thorough reading of the "Deploying Microsoft Software Update Services" guide.

Automated Analysis

Currently, there are only a couple tools available to analyze the logs. A search of the web using Google doesn't return any tools advertising the ability to analyze IIS logs for SUS activity, but a search of the microsoft.public.softwareupdatesvcs newsgroup did reveal two options by Wayne Flynn Consulting Services.[25] The first option is a web tool[26] that allows administrators to upload their log files to for an immediate web based analysis by pdxconsulting. For small sites or sites with only minimal logging information (since the upload limit is 1 MB) this can be an effective tool, provided the log information doesn't also contain anything of a sensitive nature such as information from any other Web applications (if installed) or in security conscious organizations the internal IP addressing scheme. The second tool is a locally installed web application "SUS Log Reporting System"[27] that imports the IIS' log files into a MSSQL database and generates reports from that database via the web. Again this tool is not for everyone because of its requirement for a MSSQL Server backend. For organizations that don't deploy SQL Server, the added expense and skill set required for its deployment is excessive, particularly in light of the fact that scripts could also be used to analyze these files.

While the two found tools are better than nothing they are obviously not appropriate for every group deploying SUS, but luckily the design of the logs does allow for the scripting of log analysis. Because of the predictable nature of the log entries with each item appearing even if it doesn't have a value assigned, it is relatively simple for someone to write a script in Perl or VisualBasic Script to extract the lines containing wutrack and then to process them for information. The script could do anything from breaking the information up with tabs so that it

[25] http://www.pdxconsulting.com/
[26] http://www.pdxconsulting.com/sus/
[27] http://www.pdxconsulting.com/sus_reporting.zip

could be imported to a spreadsheet or database application,[28] generating reports or adding the information to a database directly.

Maintenance

In September of 2001 Gartner suggested that companies should stop using IIS because it is "…under active attack by the vast number of virus and worm writers,"[29] and while opinions vary concerning that suggestion, it is important to keep in mind that IIS is still a frequent target of attackers. Because of this increased targeting on the part of attackers the maintenance and patching of an SUS server is still of the utmost importance, particularly in light of the potential for damage. Because SUS distributes patches to end users; an experienced hacker may try to use that to their advantage so extra caution should be used in securing the SUS box. Though digitally signing patches could go along way toward protecting an end user from malicious code placed on the SUS box, nothing is foolproof and attackers may eventually find a way to plant malicious 'patches' on the server. Likewise, and this is a far more likely scenario, an attacker with access to the server could damage or remove a patch, thereby preventing its distribution. If that patch was intended to protect users from a major vulnerability, for instance MS03-007 which was discovered when it was being actively used to exploit systems,[30] the lack of a patch could have substantial impact. Both of these scenarios highlight the importance of thoroughly maintaining and monitoring an SUS server.

SUS Maintenance is much like that of any IIS Server. It is necessary to maintain its patches, back it up regularly and keep a close eye on the logs. Patching SUS, when installed as a dedicated server, will actually be a reasonably simple task requiring little disruption. Because end users are not directly connecting to the SUS server, SUS can be patched during normal working hours without significant disruption. This, of course, changes if SUS is installed on a server that handles either other web sites or other functions.

Like all server's it is of course a good idea to back up SUS servers periodically, though the nature of SUS may not require the rigorous backup schedules associated with mission critical servers. Backups make it easy to restore a specific SUS/IIS configuration using the metabase; however there are several limitations to IIS and, by extension, SUS. The metabase is essentially IIS' equivalent of the Windows registry and "contains information about all updates provided through SUS."[31] Unfortunately, in IIS 5 (which comes with Windows 2000), it is not possible to use a backed up metabase if the entire Operating System had to be reinstalled as compared to reinstalling just IIS 5.[32] This situation is to change in Windows 2003 with IIS 6, with it being possible to reinstall the operating system and restore the IIS metabase following the

[28] A sample of this sort of simple script, written specifically for this paper, can be found in Appendix A.
[29] "Gartner: Companies should drop IIS"
[30] Sullivan
[31] "Deploying Microsoft Software Update Services," 37
[32] "HOW TO: Create a Metabase Backup in IIS"

directions provided in the Microsoft Deployment Guide.[33] As a result of these limitations for IIS 5.0, care should be taken to ensure that whichever backup solution is used is able to restore the entire machine from bare metal (i.e. a computer with blank hard drives). While it isn't normally considered a backup utility, one method of accomplishing this backup, for environments with an existing backup solution that does not provide bare metal restores, would be the use of an imaging/cloning program like Symantec Ghost,[34] which could return the computer (operating system and all) to its backed up state within minutes.

Security Effect

In evaluating SUS, it is important to keep in mind its effect upon an organization's Security. Because it makes it easier to keep critical security patches up-to-date on servers and client machine it can add significantly to a sites security. However, without the ability to install Service Packs and Office updates that increased security is limited. Additionally, for organizations that don't have any experience with IIS, the web server itself may pose a new vulnerability. With IIS' history as a commonly attacked and compromised system, the consequences of this new vulnerability can not be overlooked. In the end, organizations that follow best practices and maintain their SUS servers should see an increased level of security.

Deployment Costs

Though SUS' requirements are not particularly daunting they are noteworthy and should be kept in mind during any SUS evaluation. Because Microsoft recommends that SUS should be the only thing running on the machine,[35] deploying it means there will be another machine that has to be maintained, managed and secured. In an environment where personnel are already familiar with securing and locking down IIS, the additional cost of running SUS is primarily the price of the equipment, rack or office space, power and should a site need remote access to the SUSAdmin page and not want to run Certificate Services, an SSL certificate. However, organizations lacking IIS experience should also factor in the cost of acquiring those skills either through training or staff time spent doing research.

Besides the costs involved in deploying the server, there can also be costs associated with the deployment of a client and its associated configuration changes. Because the Automatic Client is installed automatically with Service Pack 3[36] which should be deployed anyway, the client deployment costs revolve, primarily, around the method used to distribute the configuration. If it is necessary to visit each machine and configure the client by hand the deployment costs of SUS will go up substantially, however this has to be offset by the reduced cost involved in distributing critical patches. In organizations that use

---

[33] "Deploying Microsoft Software Update Services," 38
[34] http://enterprisesecurity.symantec.com/products/products.cfm?ProductID=3&EID=0
[35] "Server Requirements and Recommendations for Installing Microsoft Software Update Services"
[36] "Software Update Services Overview," 7

Active Directory, the deployment costs are substantially reduced because the group policy can be set once and used to distribute the settings throughout an organization.

Shortcomings

During its evaluation several limitations were found in the both the SUS product itself and its documentation. The limitations revolve around its patch limitations, its inability to install different groups of patches based upon destination computer and the documentation's lack of information on securing the host operating system.

Patch Limitations

Unfortunately, SUS is a limited product that can only distribute patches to computers running Windows 2000 and higher. This limitation not only stops administrators from patching the operating systems geared toward home users like Microsoft Millennium Edition, but also older enterprise class operating systems like Microsoft Windows NT which are still employed in a number of places. Though irritating, this limitation is understandable given the fact that Microsoft has ceased "Mainstream Support" for all Windows NT products.[37]

Besides the operating systems supported, SUS is also limited in the types of patches it can distribute to end systems. SUS is limited to the operating system patches deemed critical by Microsoft and excludes service packs for both the operating system and Internet Explorer. Additionally, non-operating system patches like those for Microsoft Office which have, on occasion, been critical in nature, because they closed holes in applications like Outlook, are also impossible to install using SUS despite the fact that Microsoft has a website similar to Windows Update for updating Microsoft Office <http://office.microsoft.com/ProductUpdates/>. While SUS does reduce the amount of time spent patching systems, these limitations mean that each service pack, non-critical (by Microsoft's definition) or Office update will require installation by an organization's support staff.

Lack of Granularity

Another problem with SUS is that it is not possible to distinguish who gets what patch without a complex environment of multiple SUS servers. Using SUS is, by design, an all or nothing operation – every automatic update client will download and install every approved update. The best way to establish a more granular level of control is to run a separate SUS server for each sub-group that may have different patch requirements with each SUS server getting its content from the main Microsoft Update site. In this configuration each SUS server is run on its own as though it were a single server, with approvals being made after the updates were tested in a test environment and as desired for the client group. This configuration, however, dramatically reduces SUS' ease of administration while increasing its deployment costs. In the event that this becomes necessary,

[37] "Product Lifecycle Dates - Windows Product Family"

an organization may want to consider using multiple virtual machines located on one high end server, but this would have to be decided on a case-by-case basis based upon the organizations business and IT needs.

Host Security

One area that is particularly lacking in the documentation that accompanies SUS is the lack of discussion dealing with the host computer's security. As with any web or application server, SUS can only be as secure as its underlying operating system. Though Microsoft has taken some necessary steps to ensure the integrity of the patches installed by clients, there are no absolutes. As mentioned earlier, there is still the danger that someone with access to the SUS server could potentially alter the patches or (a more likely scenario) damage them to prevent distribution. With this in mind it is vital that administrators use recognized best practices on their SUS servers. These best practices would include limiting services, disabling file and print services (though this may not be possible if the machine is not a dedicated SUS server), disabling NetBIOS over TCP/IP (again this may not be possible in some environments), using security templates to lock down the server, using IPSec to encrypt traffic to non- web related ports and potentially doing periodic vulnerability assessments using a tool like Nessus <http://www.nessus.org>.

Conclusion

Though it is not a total patch solution, SUS provides an inexpensive method to leverage the technology found in Microsoft's Windows Update Service for the deployment of tested patches by individual organizations. A significant amount of SUS' usefulness lies not just in what it provides, away of testing critical fixes prior to deployment, but in its relative ease of use. By providing a simple and straightforward web interface, Microsoft makes SUS easy to administer with little or no training. This means that, once established, the day-to-day duties of patch approval can be handled by even a very small support staff with little free time. While SUS will in no way replace fuller featured software deployment solutions like SMS, which also features the ability to perform asset management and remote troubleshooting, it can be a welcome addition to environments were support staff have been relied upon to install critical patches. By enabling administrators to test patches before they are automatically installed on client systems, SUS can reduce the support costs of patches that have unintended consequences.

In short, with its low cost, ease of installation and relatively high level of application security, SUS is a adequate 'critical' patch management solution that can be deployed quickly by groups familiar with IIS so long as Administrators understand and work within its limitations. For groups that are less familiar with IIS, the deployment time will be a little longer; however it is still a worthwhile product since it can reduce the burden of deploying some patches.

Appendix A – Sample Script

```
#!perl

################################################################################
#                                                                              #
# SUS2Tab by John Ives                                                         #
#                                                                             #
# This is a simple script to copy all of the wutrack.bin entries from a IIS #
# log file to a new tab delimited file for easy importing to a spreadsheet  #
# or simple database program. If a log file is not named as an argument it  #
# prints out a syntax message.                                                 #
#                                                                              #
################################################################################

if (length($ARGV[0]) == 0) {
       print "SUS2Tab Syntax: perl SUS2Tab.pl <file1> <file2> <etc>\n";
} else {
       ProcessFile();
}

################################################################################
sub ProcessFile {

       #Create a new log file for the output
       open(WutrackLog, ">wutrack.tab")||
              die "Sorry, I couldn't create wutrack.tab\n";

       #Print headers into the output file
       print WutrackLog "Date\tTime\tClient IP\tClient Username\tServer IP\t",
       "Server Port\tMethod\tFile\tVersion\tPing ID\tClient\tActivity\tItem\t",
       "Device\tPlatform\tLanguage\tStatus\tError\tMessage\tProxy\tStatus\t",
       "User-Agent\n";


       while (<>) {
           if (/wutrack/) {

              #Compensate for any entries missing the V= field
              if (/wutrack.bin U/){
                     $_ =~ s/wutrack.bin U/wutrack.bin\t\tU/;
              }

              #Substitute tab characters for any spaces
               $_ =~ s/ /\t/g;

             #Breaks the status fields using tabs
             $_ =~ s/\&/\t/g;

              #Print the wutrak lines to a newfile
               print WutrackLog $_;
           }
       }

       #Close the new file
       close (WutrackLog);
}
```

Appendix B – Testing Patches

The primary reason for installing SUS is the ability to test and approve patches prior to deployment to clients; however, the details of patch testing can vary extensively between organizations and even within support groups when one is deemed particularly vital. With this in mind, the following is a brief look at the commonalities of patch testing.

When testing patches it is necessary to have a strong understanding of the computers to which the patches are to be deployed, including their hardware and software configurations as well as what the computers are being used for by the end user. With this information it is possible to quickly tell if a patch is not applicable to the intended environment or if it is essential and must be installed without extensive testing. In both scenarios a thorough reading of the Microsoft Security Bulletin should provide the adequate information for this initial assessment.

Once the patch has been evaluated for its relative importance and applicability, it should be installed to a few test machines. These test machines are generally used by individuals who are either a part of the support staff or trusted, technically savvy end users who are able to run the computers through their normal routines and document any issues that arise. Additionally, a few testers may try to stress the system to see how the patched system operates under unusual circumstances like particularly low memory or high CPU utilization scenarios.

One occasionally overlooked part of the evaluation process is the use of mailing lists like Bugtraq (http://www.securityfocus.com/archive/1) and NTBugTraq (http://www.ntbugtraq.com/). Mailing lists like this often discuss not only the discovery of new bugs, but the patches that fix them and any problems that result from patching the system. NTBugTraq has proven to be particularly useful in this area and is highly recommended.

"CERT® Advisory CA-2001-26 Nimda Worm." 25 Sept. 2001. CERT/CC. URL: http://www.cert.org/advisories/CA-2001-26.html (7 Apr. 2003).

"Deploying Microsoft Software Update Services." Jan. 2003. Microsoft. URL: http://www.microsoft.com/windows2000/windowsupdate/sus/susdeployment.asp (3 Apr. 2003).

"Gartner: Companies should drop IIS." 25 Sep. 2001. CNET News.com. URL: http://news.com.com/2100-1001-273461.html (3 Apr. 2003).

"Generating a Certificate Request File Using the Certificate Wizard in IIS 5.0." 16 July 2001. Microsoft. URL: http://support.microsoft.com/default.aspx?scid=kb;EN-US;228821 (19 Apr. 2003).

"How to Configure Automatic Updates by Using Group Policy or Registry Settings." 13 Mar. 2003. Microsoft. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q328010 (3 Apr. 2003).

"HOW TO: Create a Metabase Backup in IIS 5." 27 Oct. 2002. Microsoft. 3 Apr. 2003 URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;300672 (3 Apr. 2003).

"HOW TO: Distribute Registry Changes to Computers in Windows XP." 26 Oct. 2002. Microsoft. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;310516 (3 Apr. 2003).

"IIS Lockdown Tool." 2003. Microsoft. URL: http://www.microsoft.com/technet/security/tools/tools/locktool.asp (3 Apr. 2003).

"Port 1434 MS-SQL Worm." 28 Jan. 2003. Internet Storm Center. URL: http://isc.incidents.org/analysis.html?id=180 (3 Apr. 2003).

"Product Lifecycle Dates - Windows Product Family." 28 Mar. 2003. Microsoft. URL: http://support.microsoft.com/default.aspx?scid=fh;[ln];LifeWin (19 Apr. 2003).

"Server Requirements and Recommendations for Installing Microsoft Software Update Services" 6 Aug. 2002. Microsoft. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;322365 (3 Apr. 2003).

"Software Update Services Overview." 29 Jan. 2003. Microsoft. URL: http://www.microsoft.com/windows2000/windowsupdate/sus/susoverview.asp (19 Apr. 2003).

"Systems Management Server Product Overview." 29 Jan. 2002. Microsoft. URL: http://www.microsoft.com/smserver/evaluation/overview/default.asp (3 Apr. 2003).

"Urlscan Security Tool." 2003. Microsoft. URL: http://www.microsoft.com/technet/security/tools/tools/urlscan.asp (7 Apr. 2003).

Cottam, Don <donco@online.microsoft.com> "Re: WUTRACK.BIN: What is the V= parameter?" 15 Jan. 2003 microsoft.public.softwareupdatesvcs (24 Feb. 2003)

Fossen, Jason. "Securing Internet Information Server." Bethesda: SANS Institute. 27 Aug. 2001. 156-157

Franks, J., Et. Al. "HTTP Authentication: Basic and Digest Access Authentication." June 1999. URL: http://www.ietf.org/rfc/rfc2617.txt (19 Apr. 2003).

Smith, Jeremy. "Take control of patch deploys with Software Update Services." 18 Dec. 2002. CNETAsia. URL: http://asia.cnet.com/itmanager/tech/0,39006407,39100396,00.htm (3 Apr. 2003)

Sullivan, Bob. "U.S. military computer attacked." 19 Mar. 2003. MSNBC. URL: http://www.msnbc.com/news/886524.asp?0cv=CB10 (3 Apr. 2003).