# Global Information Assurance Certification Paper

# Patching Windows Environments Using Microsoft Software Update Services (SUS)

## Abstract

It seems like everyday that we receive notifications of a new security patch or a critical update that resolves a newly discovered vulnerability in a Microsoft product. Security patches and updates are being released faster than they can be safely deployed across the enterprise by many network managers and system administrators. Microsoft has been fully aware of this issue and is now offering their customers a powerful and an intuitive patch management utility called Software Update Services (SUS).

In this paper, I will provide a high-level overview of SUS including the patch process event flow, communication methodologies, hardware and software requirements and software configurations. Then I will highlight some of the security aspects of SUS and discuss some of the benefits and limitations of the product. I will also share with the windows community how SUS was configured and deployed in my organization. My hope is to encourage system administrators, who currently lack a patch management solution, to start taking advantage of the many features of SUS in order to regain control of windows patching in their respective organizations.

## Background

Since the September 11 terrorist attacks, many companies have made security their primary focus. The threat of information warfare and cyber terrorism is real. Corporations must continue to do their best to ensure their systems are as secure as possible in order to protect themselves from potential attacks. In the last three months alone, there were over 190 new or updated bugs reported [3]. Vulnerable systems are easy targets for attackers. Some of the costliest and most famous computer viruses were simply exploits of known system or application vulnerabilities. Viruses such as Code Red [1] and Code Red II [2] were exploits of several month old vulnerabilities in IIS (check on this). These viruses would not have spread so rapidly had system administrators applied the required patches in a timely manner shortly after they were made available by Microsoft.

In the past, system administrators had to manually check Microsoft's website for new security patches and updates, download the patches and all prerequisite patches required, test the patches, and then distribute and install the patches on each target system. Organizations without a reliable software distribution mechanism remain more at risk because they need to spend a considerable amount of time manually pushing the patches out to hundreds; maybe thousands of systems. An automated patch management solution will help save hours; maybe days over a manual approach. A few hours or a few days might not seem like much, but could be the difference between being secure and staying

vulnerable, especially after we learn of a major virus or worm is out roaming the Internet.

Fortunately, Microsoft recognized the need for an automated security patch management solution and, in June 2002 [4], released a free utility called Software Update Services (SUS version 1.0) that addresses some of the common patch management issues administrators face on a regular basis. Microsoft has since released the first service pack for SUS and plans on including some of the SUS functionality in the next version of SMS [5].

Currently, SUS only supports critical security patches and updates for Windows 2000, Windows XP, Windows Server 2003, IIS and IE. SUS does not yet support hot fixes and patches for other Microsoft products such as MS Office and SQL Server. In addition, SUS is not a true enterprise-wide patch management solution such as PatchLink [6], which can distribute patches to Unix and Netware clients as well as Windows clients. However, SUS is still an effective utility Windows system administrators can install and start using today to get a good handle on their Microsoft patch situation. Did I mention that it's also free?

**Overview of SUS Architecture**

SUS utilizes a client-server architecture, which extends the functionality of the existing Windows Automatic Updates technology. The Automatic Updates (AU) client was first shipped with Windows XP service pack 1 and SUS builds on that technology. The AU client was originally designed to communicate directly with a Microsoft update server over the internet and was geared towards securing user's home PC's running Windows XP.

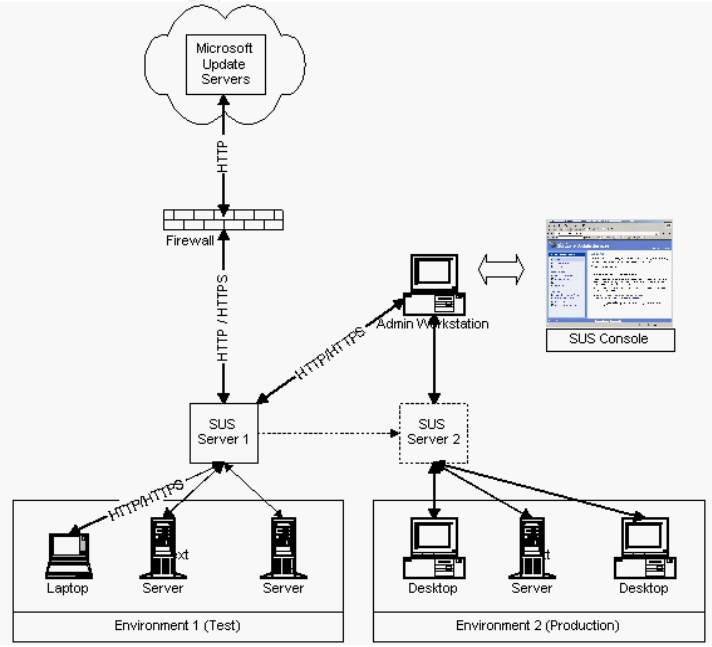


Figure 1 – SUS Architecture 1

Rather than download patches directly from Microsoft's windows update servers, the same AU client can now be configured to connect to a local SUS server. On the other hand, the SUS server maintains a current list of patch descriptions (metadata) and a repository of the actual patches (executables and .cab files). This is accomplished by synchronizing with Microsoft's windows update servers at an administrator-defined interval.

The SUS architecture also allows for multiple SUS servers to coexist in the same environment, and therefore, scales well for large networks. Multiple SUS servers can be deployed and strategically placed across the enterprise to support satellite offices, remote sites, and DMZ environments. A least one primary SUS server needs to be configured to download patches directly from the windows update servers. The remaining subordinate SUS servers, if any, can be configured to synchronize with the primary SUS server. Having one primary SUS server, allows the system administrators to manage all incoming patches from a central location, thus preventing any duplication of effort and reducing the effort needed to support multiple primary SUS servers. However, this is not a software requirement and the administrator may wish to setup additional primary SUS servers if so desired.

### *Patch Process Flow*

The following flowchart [7] created by Peter Pawlak describes how the SUS patching process works and illustrates the sequence of events from initial patch download to final installation.

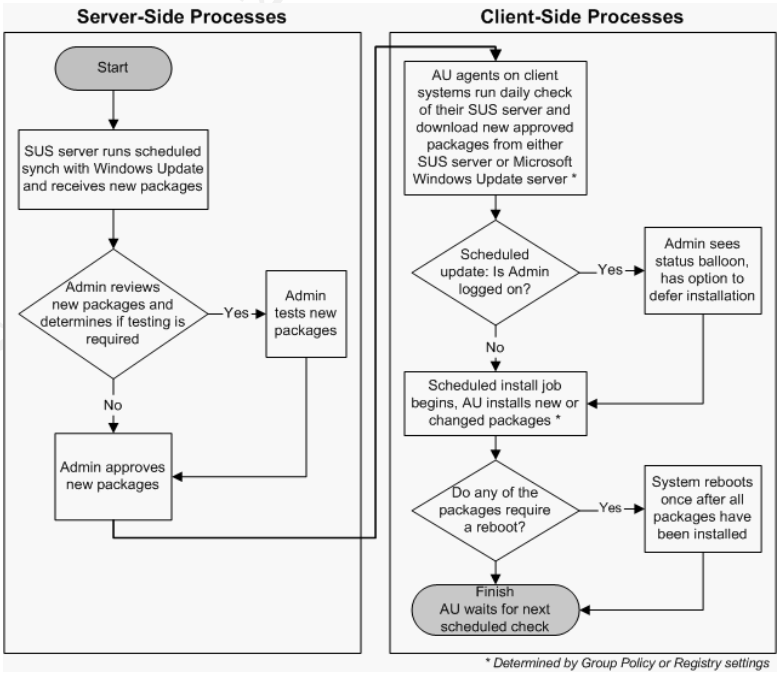


Figure 2 – SUS Patch Process Flowchart

On the SUS server, the downloaded patches can only be approved by authorized system administrators with local administrative rights on the server. Patches should never be approved without extensive testing in an integration test environment that closely mirrors production. Once a patch has been approved, the AU clients are then allowed to download and then later install the patch.

***SUS Server Requirements & Installation Notes***

The latest version of SUS server (1.0 with SP1) must, at a minimum, meet the following hardware and software requirements:

| **Hardware Requirements** | |
|---|---|
| CPU | Pentium III 700 MHz |
| Memory | 512 MB RAM |
| Disk Space | 6 GB |
| **Software Requirements** | |
| Operating System | - Windows 2000 service pack 2 or higher *<br>- Windows 2003 * |
| Web Browser | Internet Explorer 5.5 or higher |
| Web Server | IIS 5.0 or higher |
| Number of AU clients | < 15,000 per each SUS server |
| * Server cannot be an Active Directory Domain Controller | |

Table 1 – SUS Server HW & SW Requirements

The installation of the SUS server is straightforward and only takes a few minutes to complete. The SUS server software can be downloaded from Microsoft's web site at the following URL:

http://www.microsoft.com/downloads/details.aspx?FamilyId=A7AA96E4-6E41-4F54-972C-AE66A4E4BF6C&displaylang=en

The installation steps are clearly outlined in the document titled "Deploying Microsoft Software Update Services" [8] and are outside the scope of this document. However, I will attempt to highlight a few notes about the installation process I feel might benefit those reading this document.

The SUS server is virtually identical to Microsoft's windows update server and can be best described as an intranet-version (local copy) of the windows update servers. At the core of the SUS sever, is Microsoft IIS 5.0. Before installation, it is recommended that the target SUS server be patched with the latest patches. Obviously, you don't want your enterprise patch server to be un-patched!

It is important to note that the installation of SUS server performs the following actions automatically:
1. Installs Microsoft IIS version 5.0 web server, if not already installed.
2. Installs the IIS lockdown tool version 2.0, if not already installed.

3. Installs the IIS URL Scanner version 2.5, if not already installed.

Once the installation is complete, you'll have a hardened, airtight web server out-of-the-box.

### *Automatic Updates Client Requirements & Installation Notes*

The AU client is currently supported on the following Windows operating systems:

| Operating System | Service Pack |
|---|---|
| Windows XP Home & Professional | - |
| Windows 2000 Professional, Server, and Advanced Server | 2 |
| Windows 2003 (.Net) | - |

Table 2 – AU Client Operating System Requirements

The AU client was initially bundled into Windows XP service pack 1, then in Windows 2000 service pack 3. Microsoft will also include the AU client in the Windows 2003 (.Net) family of operating systems. Organizations that have not yet deployed Windows XP service pack 1 or Windows 2000 service pack 3 across their organization can download the AU client software directly from Microsoft's web site at the following URL:

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

It is also important to note here that the deployment and installation of the AU clients will most likely consume the most amount of time throughout the SUS deployment effort. Therefore, I strongly encouraged system administrators to device creative ways to mass-distribute and install the AU client software across their enterprise. One possible method would be to use windows logon scripts, or maybe distribute the AU client software using your organization's well-established software distribution mechanism (if one exists). The good news is that, once installed, the AU client can be administered remotely fairly easily.

### *SUS Server Communication Methodology*

The primary SUS server communicates with the Microsoft windows update servers over HTTP port 80. No authentication is done against Microsoft's update servers. However, SUS checks the digital signatures of the patches to ensure they are authentic before downloading them locally. Communication between the primary SUS server and the subordinate SUS server(s) is either HTTP over port 80 or HTTPS over port 443. By default, SUS server communication is over port 80. I recommend that administrators invest the time to configure SUS to use signed SSL certificates. The process is identical to creating internal SSL certificates for IIS. The steps on how to do that are clearly

outlined in the "Deploying Microsoft Software Update Services" [8] guide and are outside the scope of this paper.

Internally, the subordinate SUS servers communicate with the primary SUS server over HTTP or HTTPS. All subordinate SUS servers should also be configured to use SSL if the primary server is configured to do so. This ensures all SUS communication is 128-bit encrypted at all times.

### *Automatic Updates Client Communication Methodology*

The AU client utilizes the Binary Intelligent Transfer Service [9] (BITS) to download patches from the SUS server using the HTTP protocol. BITS is a proprietary file transfer service from Microsoft that has two distinct benefits: 1) It throttles the download to limit network impact, and 2) it can recover from an interrupted download. The AU client is able to determine which HTTP protocol to use from the URL address of the SUS server. For example, if the client connects to https://sus-server1 then it will communicate over port 443.

One important thing to note here is that the SUS server does not attempt to push patches out to the AU clients. Rather, each AU client queries the SUS server for newly approved patches and then pulls those patches down locally. In other words, SUS patch deployment is a pull operation from the clients rather than a push operation from the SUS server. This deployment methodology differs from other enterprise software distribution tools, such as Tivoli Software Distribution, which pushes out software from a centrally located software repository server. Arguing which distribution mechanism (pull versus push) is better can start a heated debate among enterprise system managers. However, the pull methodology has these following two benefits:

1. Greatly increases the distribution success rate since the target system is almost guaranteed to be "up".
2. Simplifies the overall architecture because intermediary distribution servers that 'cache' the software are not necessary.

### *SUS Server Configuration*

The management and configuration of the SUS server is entirely web-based and can be accessed by simply browsing to: http://sus-server1/susadmin. The server administration interface is a simple point-and-click web interface that is extremely easy to use and navigate.

Those of us familiar with the Windows Update website will find the SUS administration interface very similar and just as useful. The 'susadmin' page also contains links to the windows update web site as well as other useful links to the Microsoft knowledge base and the Microsoft security site. Below is a screen capture of the SUS welcome screen. Look familiar?

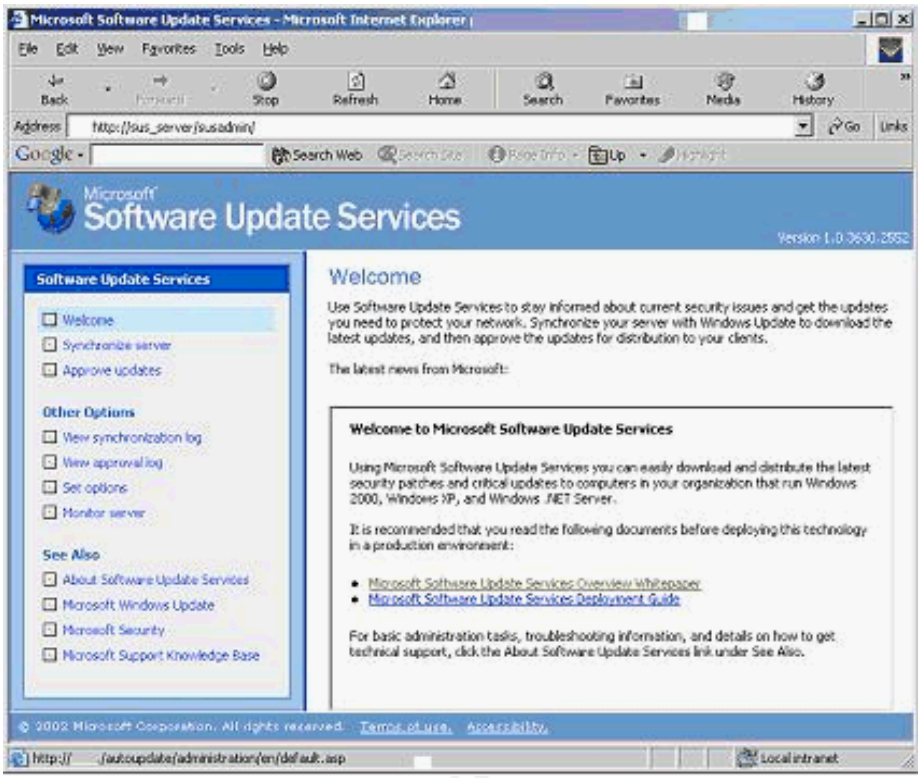


Figure 3 – SUS Welcome Screen

Upon the successful installation of SUS, administrators must logon to the SUS server by browsing to http://<sus-server-name>/susadmin, and then perform the following configuration steps:

1. Select a proxy server configuration. This must be setup first to allow the SUS server to connect to the Internet and download the patches from Microsoft.
2. Specify the NetBIOS name of the SUS server. The fully qualified domain name of the SUS server or its IP address can also be used.
3. Select which server to synchronize content from. This is where you can configure the SUS server as a primary or as a subordinate server. If you select the option to synchronize directly from the Microsoft update servers then the SUS server becomes a primary.
4. Select how to handle updates of previously approved patches. Often times, a newly released patch may actually cause more damage and may not exactly work as advertised. In this case, Microsoft may release an update for that patch. You have the option to allow SUS to automatically approve a previously approved patch or do it manually. I recommend setting it to manual approval for additional control.
5. Select where patch content is to be stored. Patch content can either be stored locally on the SUS server or maintained on Microsoft's update servers. If disk space on your SUS server is an issue then you might want to select the option to maintain the patches on Microsoft's server. Otherwise, select the option to store the patches locally.

6. Manually synchronize with Microsoft's update servers. The time for the initial synchronization will be the longest and depends on the speed of your Internet connection. The initial synchronization in my organization lasted about 31 minutes over a T1 line. The number of patches and updates that were downloaded are shown in the table below:

| Microsoft Product | Number of Updates |
|---|---|
| Internet Explorer 5.0X | 96 |
| Internet Explorer 5.5X | 144 |
| Internet Explorer 6.X | 225 |
| Windows 2000 | 847 |
| Windows XP | 770 |

Table 3 – Security Patches and Updates Currently Available

7. Configure the synchronization schedule. This tells the SUS server how often it needs to synchronize patch content. The synchronization schedule should be adjusted to synchronize content during off-peak hours to minimize impact to the network. Microsoft recommends that you synchronize once a day.

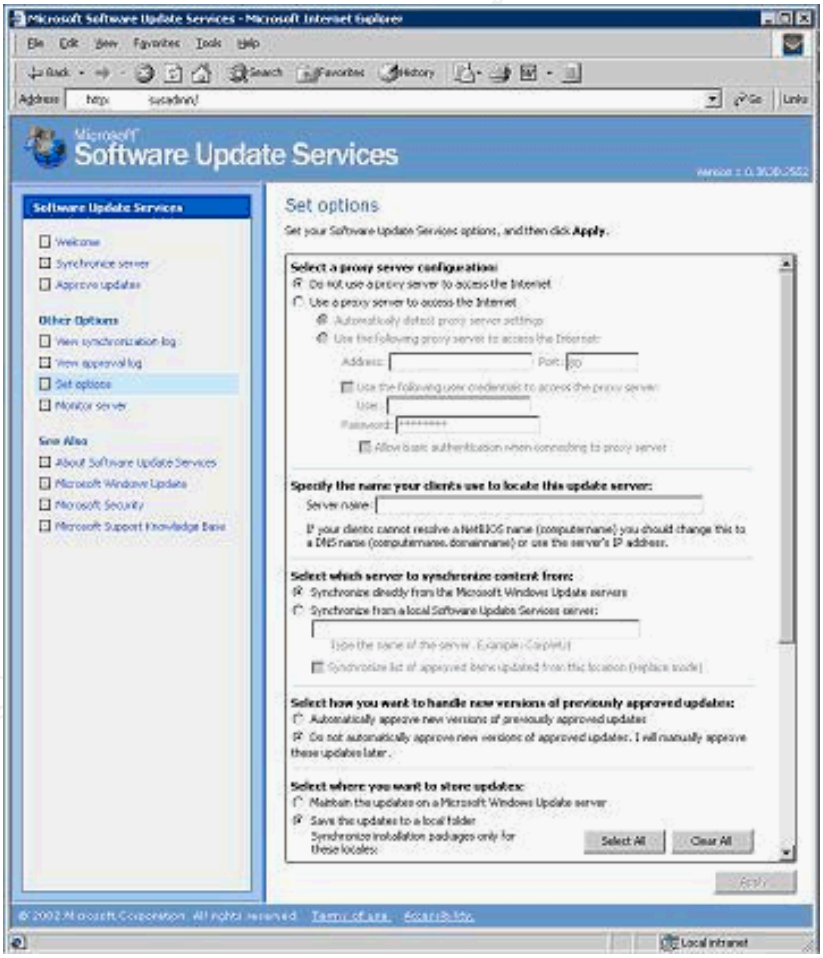


Figure 4 – SUS Setup Options

### *AU Client Configuration*

The AU client on the client can be configured in one of two ways:

1. Locally from the client. A local administrator can launch the Automatic Updates configuration screen and make the changes locally. On Windows XP, the Automatic Updates configuration screen can be accessed through System Properties, then selecting the 'Automatic Updates' tab. On Windows 2000, the Automatic Updates configuration can be accessed through the Control Panel.

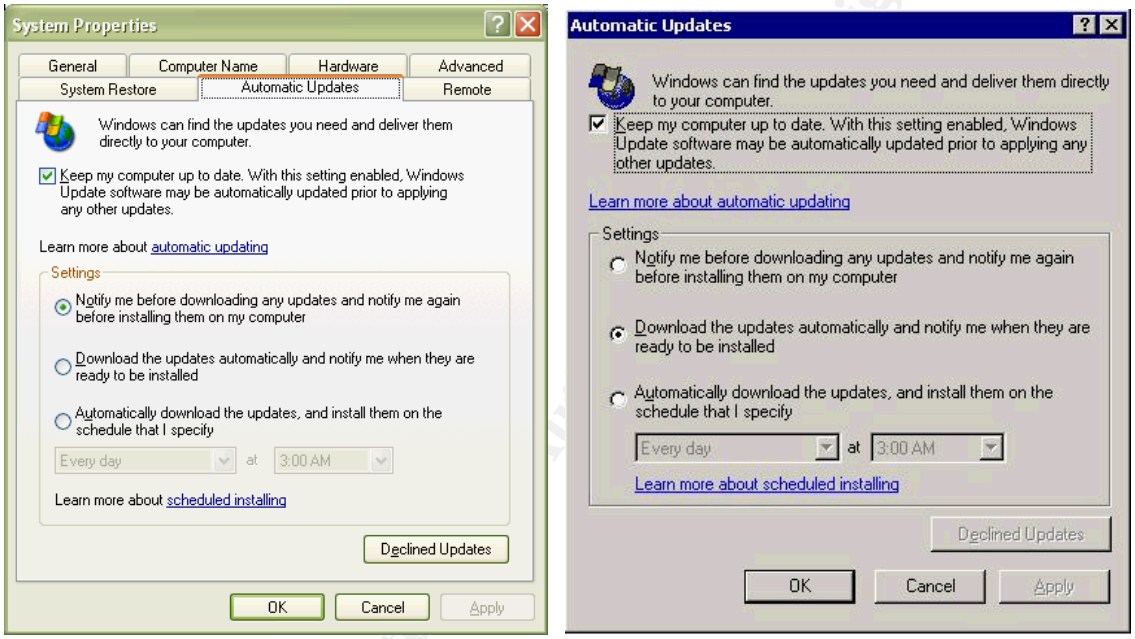


Figure 5 – AU Client configuration for Windows XP and Win 2K Respectively

2. Via a remote registry edit. This option should be strongly considered by administrators wishing to configure the AU client on a large number of systems in their environments without the use of a software distribution mechanism. The following registry keys need to be configured under:

   HKLM\Software\Policies\Microsoft\Windows\WindowsUpdate\AU

| Registry Key | Registry Type | Possible Values |
|---|---|---|
| RescheduleWaitTime | REG_DWORD | 1-60 (wait time in minutes) |
| NoAutoRebootWithLoggedOnUsers | REG_DWORD | 0-1, 1 = give user option to postpone reboot |
| NoAutoUpdate | REG_DWORD | 0-1, 1 = enabled |
| AUOptions | REG_DWORD | 2,3,4 (download settings) |
| ScheduledInstallDay | REG_DWORD | 0-7 (day of week) |
| ScheduledInstallTime | REG_DWORD | 0-23 (hour of day) |

| | | |
|---|---|---|
| UseWUServer | REG_DWORD | 0-1, 1 = enabled |
| WUServer | REG_SZ | http://sus-server1 |
| WUStatusServer | REG_SZ | http://sus-server1 |

Table 4 – AU Client Registry Keys and Values

### Software Update Services Security Features

As we have seen thus far, SUS has some great built-in security features that make it a serious contender in the windows Patch Management arena. Below is a summary of those strengths:

1. SUS server downloads verified Microsoft patches only after ensuring that the patch signatures are authentic. Patches that don't pass this criterion are ignored.
2. Supports HTTPS communication for secure data transmission and administration over the network. Data communication between the different SUS servers, as well as from AU clients to SUS server(s), can be easily encrypted using 128-bit SSL encryption.
3. IIS lockdown tool 2.0 and URL scanner 2.5 are installed automatically.
4. Patches can only be approved by authorized system administrators.
5. Strong auditing capabilities. Synchronization activity and patch approval history is logged and can be saved in multiple different formats.
6. The latest version of the Microsoft Baseline Security Analyzer (MBSA version 1.1) has built-in support for SUS. MBSA can now compare installed patches on different computers system against approved patches on the SUS server and generate reports on any critical patches missing.

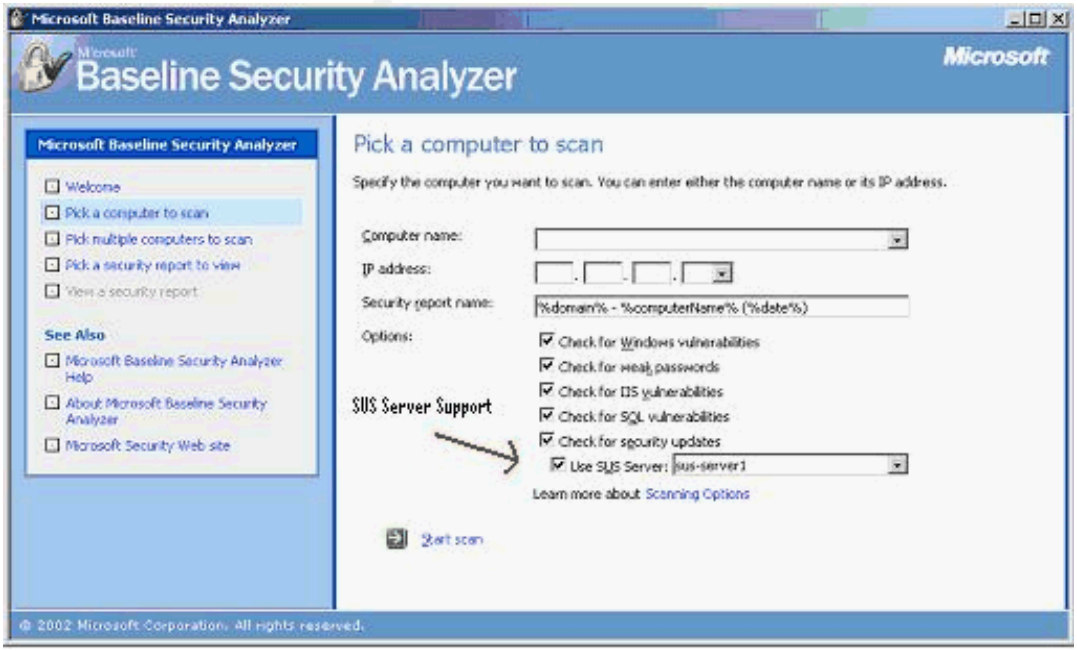


Figure 6 – Microsoft Baseline Security Analyzer Now Supports SUS

## Benefits & Limitations

I hope by now you were able to see some of the major benefits SUS has to offer. But let's not fool ourselves. SUS also has limitations just like any other application. I have summarized the benefits and limitations of SUS below.

### *Benefits*

- Intuitive interface makes the product very easy to use and manage.
- Able to handle critical security patches and updates for Windows 2000, Windows XP, Windows 2003 Server, IIS, and IE. This list is expected to increase to include support for Office and SQL server patches.
- One SUS server can handle up to 15,000 AU clients.
- Scalable. Additional SUS servers can be deployed as needed.
- Cost-effective and time-effective patch management.
- SUS server installation is secure out-of-the-box.
- Powerful auditing and logging capabilities.
- It's free!

### *Limitations*

- Automatic Updates client software must be distributed and installed on each system wishing to receive patch updates via SUS.
- Patch approval is an all-or-none process. SUS currently lacks the granularity to approve different sets of patches for different groups of clients.
- Automatic Updates client cannot be uninstalled after installation.
- SUS server uninstall does not delete changes made by the IIS lockdown tool.
- SUS does not support general hot fixes.
- SUS does not support patches for Microsoft Office and SQL products.
- SUS does not support patches for Windows 95, 98 or Windows ME.
- No automated notification of when new critical patches have been downloaded. Administrator must manually check the synchronization log.

## How SUS Was Configured and Deployed in my Organization

### *The Requirement*

About a year ago, the Information Services department was tasked with selecting a COTS (commercial of-the-shelf) product that can help us quickly and efficiently deploy Microsoft service packs and critical security patches. Our organization has had its fair share of virus attacks and has suffered downtime as a result of systems in our environment not being patched. In fact, at one point, it was thought that as long as we have firewalls protecting our borders we were secure. But viruses such as Code Red and Nimda quickly proved us wrong. The initial scope was to select a do-it-all patching product that manages patches for Unix systems as well as Windows servers and Desktops. But after further research, it was determined that a full-fledged patch management product would cost more

than what we wanted to spend—and since our systems are 99% Windows, selecting SUS as the winner was a very simple decision. It was free and it did what we needed it to do.

### *The Production & Test Environments*

The production environment consists of the following systems:

| System Type | Operating System | Number of Systems |
|---|---|---|
| Unix Servers | Sun Solaris 8 | 24 |
| Windows Servers | Windows 2000, SP2 | 116 |
| Workstations | Windows 2000, SP2 | 2850 |

Table 5 – Summary of Production Systems

As you can see above, the production environment has close to about 3000 nodes, mostly running Windows 2000. The production network is segmented into three physically separate networks and protected by multi-layer firewalls. The test environment closely resembles the production environment and contains a small subset of the production servers.

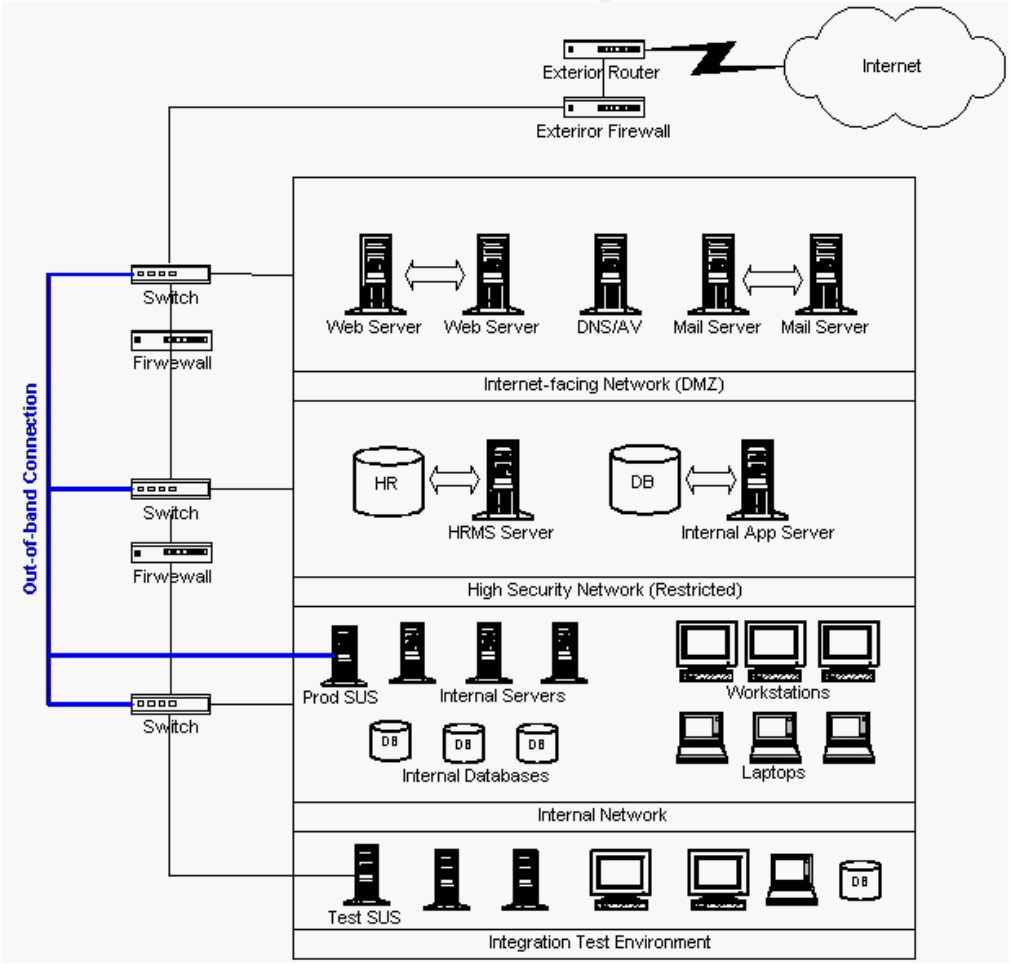


Figure 7 – High Level Network Diagram

Fortunately, my company has a very strict change control process. In general, we do not allow installation of new software or hardware into the production environment without us first rigorously testing in the test environment. Having a good test environment has allowed us to minimize downtime and drastically reduce impact to production operations.

### *The Installation & Setup Process*

Initially, SUS was installed in the test environment on a server running Windows 2000 service pack 3. Having service pack 3 installed first was important because it included the AU client software. Once the AU client was configured, we were able to connect to the windows update server and install the latest critical patches and updates from Microsoft.

We also decided to make the SUS server in the test environment become our primary SUS server. Downloading the patch content from the Microsoft update servers directly into the test environment made a lot of sense since that was where patch testing was going to take place.

Later on, another SUS server was installed and configured in production. In keeping with our strict change control processes, we ensured that the production SUS server was built identically and contained the same patches as the SUS server in the test environment. The production SUS server was then configured to synchronize content with the test SUS server. The key here is that only approved patches are synchronized. This ensured that we did not have any non-approved patches in our production environment.

### *Network Placement*

As you might be able to see from the network diagram in figure 7, the production SUS server was placed on the internal network segment. This made accessing the server and managing it a little easier. Since we have a fairly small number of nodes (just under 3000), it did not make sense to deploy a separate SUS server for each physical network (DMZ, secured, and internal). Frankly, our budget did not allow for the purchase of additional server hardware. Therefore, two additional network cards were installed in the production SUS server, for a total of three. Each network interface card was configured with an IP address that is within the range of the network segment it was connected to. The network connections were out-of-band and did not go through our standard switched network. In other word, each NIC was directly connected to the switch for that network segment. From the network perspective, it appears as though there are three separate SUS servers. The AU clients in production don't know that they are all getting their patch updates from one SUS server. Brilliant!

We have been using this configuration for over six months now with great success. We have not had any issues synchronizing patches or distributing patches to the AU clients. Below is a snapshot of the NIC configuration on the production SUS server.

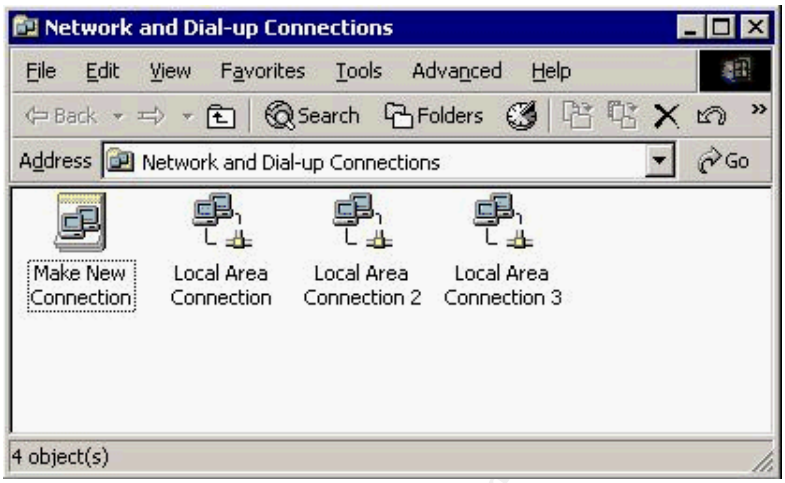


Figure 8 – SUS Production Server NIC Configuration

### *Hints & Lessons Learned*

1. Make sure you have a good test plan for testing patches. Involve the proper support staff to ensure patches don't introduce any new issues.
2. Make sure you have a good backout plan. Patches installed on client systems using SUS can be uninstalled using Add/Remove program.
3. It is a good idea to update your standard OS builds (gold builds) once new service packs and critical patches have been approved and successfully tested. Ensure new servers and workstations use the new build.
4. Don't manually synchronize during peek business hours. Always schedule the synchronizations during off-peak hours. Make sure you take into consideration the hours of the night when system backups are running.
5. Perform the AU client installs during off-peak hours to minimize network impact.

### Conclusion

"Applying patches is like eating vegetables or exercising: you may not like it, but you've got to do it to stay healthy" [8]. I have to agree completely with that statement. Patch management is a tedious and time-consuming process. Nobody wants to do it. Fortunately, SUS offers a wealth of patch management features and is freely available. If you have struggled in the past with service packs and critical updates then worry no longer. SUS might just be the only hope you have to stay afloat when the next patch tidal wave hits. And with new versions of SUS coming out soon, it's only going to get better. So what are you waiting for? You can download it and start using it today.

## References

[1] Symantec Corporation, "CodeRed Worm." September 24, 2002
URL: http://www.symantec.com/avcenter/venc/data/codered.worm.html

[2] Symantec Corporation, "CodeRed II." October 29, 2002
URL: http://www.symantec.com/avcenter/venc/data/codered.ii.html

[3] Security BugWare, "Bugs for the last three months." February 19, 2003
URL: http://www.securitybugware.org/new.html

[4] InfiniSource Corporation, "Microsoft Releases Software Update Services." June 24, 2002.
URL: http://www.windows-help.net/microsoft/sus.html

[5] Microsoft Corporation, "Software Update Services Overview Whitepaper." June 2002.
URL: http://www.microsoft.com/windows2000/docs/SUSOverview.doc

[6] Conry-Murray, Andrew. "PatchLink Update 4.0 and BigFix Enterprise Suite 2.0." February, 5, 2003.
URL: http://www.networkmagazine.com/article/NMG20030205S0005

[7] Pawlak, Peter. "Software Update Services Flowchart (Illustration)." April 22, 2002
URL:http://www.directionsonmicrosoft.com/sample/DOMIS/update/2002/05may/0502sustep_illo1.htm

[8] Microsoft Corporation, "Deploying Microsoft Software Update Services." January, 2003.
URL: http://www.microsoft.com/windows2000/docs/SUS_Deployguide_sp1.doc

[9] Pawlak, Peter. "What is BITS (sidebar)." April 22, 2002
URL:http://www.directionsonmicrosoft.com/sample/DOMIS/update/2002/05may/0502sustep_sb.htm

## Other Useful Sources

Microsoft Windows Update Website: http://windowsupdate.microsoft.com/

Microsoft Security Web site: http://www.microsoft.com/security/

PatchLink web site: http://www.patchlink.com/

Microsoft SUS Frequently Asked Questions:
http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/ittasks/support/corpwu.asp

Microsoft SUS Server Software:
http://www.microsoft.com/downloads/details.aspx?FamilyId=A7AA96E4-6E41-4F54-972C-AE66A4E4BF6C&displaylang=en

Microsoft Automatic Update Client Software:
http://www.microsoft.com/windows2000/downloads/recommended/susclient/default.asp