Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Case Study:Use Caution When Deploying Microsoft's Software Update

Microsoft has quietly developed the Software Update Service (SUS) for distributing critical software updates and patches. Once installed, and properly configured, an internal SUS website will respond to internal hosts requesting the latest operating system patch or security roll-up, just like the Windows Update website. The purpose of this case study is to document the process used to evaluate the security risks associated with SUS before implementing it on a real world network. Risks such as hardening IIS, protecting ...

Copyright SANS Institute

Author Retains Full Rights

AD

Case Study: Use Caution When Deploying Microsoft's Software Update Services On A Small Network
November 10, 2002
James McVicar

Abstract

Microsoft has quietly developed the Software Update Service (SUS) for distributing critical software updates and patches. Once installed, and properly configured, an internal SUS website will respond to internal hosts requesting the latest operating system patch or security roll-up, just like the Windows Update website. The purpose of this case study is to document the process used to evaluate the security risks associated with SUS before implementing it on a *real world* network. Risks such as hardening IIS, protecting the Internet connection required when downloading updates from the Internet, and server placement within the network were considered. Ultimately, I hope to demonstrate how I used Microsoft's Software Update Services as a solution for delivering the latest operating system updates and patches to internal clients on a small network. (WARNING: A recently discovered vulnerability may make this product extremely unsafe if configured incorrectly. Suggested configuration changes are noted in this paper)

Introduction

My 'real world' problem

Recently, my company acquired the responsibility of providing infrastructure and technical support to a small, non-profit organization. (See Appendix A for network diagram) Shortly after acquiring this responsibility, I experienced a 'major' problem. Several end users had reported application errors after applying the latest Internet Explorer 5.5 update found on the Windows Update website (http://windowsupdate.microsoft.com). After a lengthy investigation, it was discovered that the update had modified a Windows system file, (gdi.exe) causing a protection fault whenever a user attempted to print from the company's financial software application (Solomon v2.6). In effect, the update acted like a virus, disabling the company's most important application. The company experienced a loss of production in addition to the expenses incurred while diagnosing and repairing the problem. While investigating the problem, I uncovered the following security related issues. First, workstations were randomly updated; some were updated while others were not. I used Microsoft's patch checking utility (hfnetchk.exe) to document the discrepancy between workstations that should have been equally patched. (See Appendix C for how I used hfnetchk.exe to compare patches on workstations) Second, end users would *'blindly'* apply any patch found on the Windows Update website. According to *Peter Pawlak*'s, web article, "Software Update Service to Ease Patch Distribution",[i] there are four basic strategies companies use for delivering software patches to their workstations:

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

1. Do nothing.
2. Have qualified personnel manually install the patches.
3. Allow end users to update their own workstations.
4. Use a software distribution product to deliver patches.

This *real world* company relied on the first, and third, of Pawlak's strategies when applying security patches culminating with end users unwittingly crashing an important business application.

Applying security updates is an important security task; the SANS Institute recognized that failing to install security updates is ranked among the top 5 worst security mistakes made by end users. [iii] Unfortunately, end users from this company were not qualified to determine when, and which patches to apply, therefore I was tasked with finding a solution that would to address the issue of consistently applying security updates to all hosts on their small network. My solution had to have a method of delivering the latest patches to every workstation, and file server, while preventing end users from installing unnecessary patches. Furthermore, the solution had to be inexpensive and easy to deploy. I decided to implement Microsoft's Software Update Services. When compared to more costly and technically advanced solutions like Microsoft's SMS (http://www.microsoft.com/smserver/default.asp) or Patchlink Update 4.0 (http://www.patchlink.com), SUS was an appropriate choice for this small network.

A closer look at Microsoft's Software Update Services

The Automatic Update service

Microsoft's Software Update Service solution consists of two components; an Automatic Update (AU) client agent, used to request operating system updates, and an IIS website, used to configure the service and respond to requests made from AU clients. The client agent runs as a Windows service and is included with Windows 2000 Service Pack 3, Windows XP Service Pack 1, and .NET servers, or may be downloaded (WUAU22.msi) as a separate component. The service pack will automatically install the client; otherwise you must launch the WUAU22.msi file to install the service. Either method requires further configuration in order to direct the AU service to use an internal SUS server rather than the default, Microsoft Windows Update website. The AU client lacks a GUI and is therefore cumbersome to configure since you must modify the registry using Active Directory Group Policies, or manually *hack* the local registry in order to force AU service to use an internal SUS server. Once configured, the settings can be easily transferred to other workstations using an Active Directory Group Policy or manually distributing, and applying, a Registry script file. You could use a variety of software deployment methods to distribute the script in a non-Active Directory network. I used a script, (See Example 1) in my test environment, to configure the AU service to automatically my poll internal SUS server every day at 01:00 and automatically install the update. You can modify

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

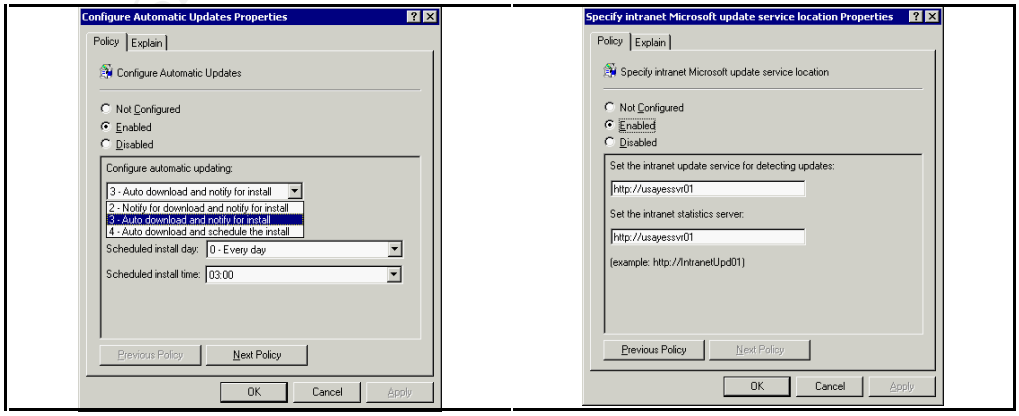
the AUOptions value to determine if the process is automatic, or if it requires an end user response. Once configured, the AU service runs under the context of LocalSystem, but can be changed to run under the context of another user. Windows uses the LocalSystem account to run services regardless of who is logged on. While this could present a security risk in some cases, allowing the AU service to run as LocalSystem was acceptable in my *real world* environment because I wanted the patches to be applied regardless of who was logged on. My *real world* network allowed me to use the Active Directory Group Policy method to apply the settings to domain computers. This required using the Group Policy editor and the wuau.adm policy template (installed by SUS). Example 2 shows the screens used by the Group Policy editor. The Microsoft "Deploying Microsoft Software Update Services" guide outlines this process in detail. [vi] (pgs. 55-59)

Example 1: Registry script used to modify the Automatic Update service on client workstations

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate]
"WUServer"="http://yourSUSserver"
"WUStatusServer"="http://yourSUSserver"

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU]
"UseWUServer"=dword:00000001
"NoAutoUpdate"=dword:00000000
"AUOptions"=dword:00000004
"ScheduledInstallDay"=dword:00000000
"ScheduledInstallTime"=dword:00000001
```

NOTE:
Copy the above text into a file called *filename.*reg (example: auupdate.reg)
Double click the file to update the local registry.
WUServer: IP address or DNS name of your server running SUS
WUStatusServer: IP address or DNS name of your server running SUS
AUOptions: 2=notify before download and install, 3=notify before install, 4=automatic install
ScheduledInstallDay: 0=everyday, 1-7=single day of week
ScheduledInstallTime: 0-23=24 hour format

Example 2: Using Group Policy editor to modify the Automatic Update service on client workstations



© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Once configured, the clients will automatically begin to make requests. If it is determined that a download is required, the AU service will hand the download process to another Windows service called BITS. BITS (Binary Intelligent Transfer Service) must run under the context of LocalSystem in order for the Automatic Update service to proceed automatically, otherwise BITS will suspend the download process whenever the user logs out. [iv] (pg. 6) By default, the service is set to Manual and will start whenever a service requests a download. The SUS server is configured to respond, and deliver patches using BITS technology. [vii] (pg. 48) Microsoft introduced this service as a method of throttling bandwidth during file transfers since BITS will adjust the download speed to avoid capitalizing the local system's network resources. For a detailed explanation of this service, please see Microsoft's, Binary Intelligent Transfer Service white paper. [iv]

The Software Update Services website

The main component of Microsoft's Software Update Services is the website, and can be downloaded (SUSsetup.msi) from Microsoft for no charge. The website serves two purposes. First, the website contains administrative pages used to configure SUS. Second, the website locally stores the update content for distribution to AU clients. This website must be installed on a Windows 2000 server running Microsoft's IIS 5.0. Microsoft recommends that no other websites be hosted on the SUS web server. [vii] (pg. 9) If you install SUS on an empty IIS server, it will create its own website on port 80, otherwise SUS will install the administrative pages in whatever site is already bound to port 80. [vii] (pg. 64) One author, Bret Hill (iisanswers.com), cautions against installing SUS on existing IIS webservers since many components, like the Metabase and urlscan.ini, are overwritten without notification. [v] The SUS installation automatically runs the IIS Lockdown utility which removes or disables many services and directories. Most notable are: disabling the **FTP**, **SMTP** and **htr scripting** services. It also removes the **scripts**, **iisamples**, **MSADC**, and **iisadmin** virtual directories and disables the Anonymous user **execute system utility** and **content directory** write abilities. A detailed list of disabled services is well documented in the deployment guide. [vii] (pg. 66-67)

Once installed, administrators must decide how the SUS should contact the Windows Update website and, how the updates should be distributed. Configuration parameters include, but are not limited to:

- Should the SUS service manually or automatically synchronize with Microsoft's Windows Update website?
- Once synchronized, should the updates be manually or automatically approved before distribution to AU clients?

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

- Will the SUS server distribute the updates, or should AU clients be redirected to the Microsoft Windows Update website?

This process is handled using administrative web pages that are intuitive and easy to navigate. The web pages are served through standard HTTP:80 requests, but can be configured to use Secure Socket Layers (SSL). However, SSL requires purchasing and installing a digital certificate which was not necessary for my *real world* environment.

The SUS service may be configured (See Appendix D) to store the updates and patches on the local server. (NOT RECOMMENDED! Due to a recently discovered vulnerability, I suggest redirecting your AU clients to the Microsoft Windows Update website. This will allow you to control which updates your clients may apply, but the content is distributed directly from Microsoft. This will require an Internet connection for your AU clients) If the content is stored on the server, it's in a folder (%SUSInstallFoder%/content/cabs) with the following NTFS permissions. (See Example 3) The website uses IIS virtual directories to access the content in this folder. Notice how the Web Applications group and Anonymous Users are specifically denied the ability to write, append or modify the data in this folder. Otherwise, the folder would have inherited permissions associated with the Everyone group, and would be easier to exploit.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Example 3: NTFS permissions on ..\cabs folder

```
c:\sus\content\cabs USAYESSVR01\Web Applications:(OI)(CI)(DENY)(special access:)
                    DELETE
                    WRITE_DAC
                    WRITE_OWNER
                    FILE_WRITE_DATA
                    FILE_APPEND_DATA
                    FILE_WRITE_EA
                    FILE_DELETE_CHILD
                    FILE_WRITE_ATTRIBUTES
          USAYESSVR01\Web Anonymous Users:(OI)(CI)(DENY)(special access:)
                    DELETE
                    WRITE_DAC
                    WRITE_OWNER
                    FILE_WRITE_DATA
                    FILE_APPEND_DATA
                    FILE_WRITE_EA
                    FILE_DELETE_CHILD
                    FILE_WRITE_ATTRIBUTES
          Everyone:(OI)(CI)(special access:)
                    READ_CONTROL
                    SYNCHRONIZE
                    FILE_GENERIC_READ
                    FILE_READ_DATA
                    FILE_READ_EA
                    FILE_READ_ATTRIBUTES
          BUILTIN\Administrators:(OI)(CI)F
          NT AUTHORITY\SYSTEM:(OI)(CI)F
          Everyone:(OI)(CI)(special access:)
                    READ_CONTROL
                    SYNCHRONIZE
                    FILE_GENERIC_READ
                    FILE_READ_DATA
                    FILE_READ_EA
                    FILE_READ_ATTRIBUTES
          BUILTIN\Administrators:(OI)(CI)F
          NT AUTHORITY\SYSTEM:(OI)(CI)F
```

NOTE: This output was obtained using a batch file. See Appendix E for source code

When the synchronization process is complete, updates must be approved for distribution. Administrators can allow SUS to automatically approve and distribute the updates or hold them pending manual approval. This is done using the administrator web pages. Since my *real world* solution had to prevent end users from *'blindly applying untested patches'*, I chose to manually approve the updates. The delay would allow me to test the patch on various client configurations before releasing them for distribution.

Unfortunately, there are a couple of significant limitations associated with this process. First, if an update is approved, and an AU client has already downloaded the update, it must be manually uninstalled at the client machine. You cannot "UN-approve" an update on the SUS server and expect the client to be forced (or notified) to remove the update. Second, patches are approved globally and cannot be isolated to a particular machine or user group. This could be a significant problem in 'real world' environments containing vastly different end-user or machine roles. My *real world* network is homogeneous, where end user roles, hardware and software environments are similar on every machine.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Using Logs to monitor and troubleshoot SUS

Troubleshooting and monitoring SUS activity requires searching through various logs. Some are available using the administrative web pages, while others must be searched for. The approval and synchronization logs (Examples 4 and 5) are easily located in the SUS administrative website main navigational menu and provide helpful information. I used the approval log to document and provide a baseline for the minimum patch level for my *real world* hosts. The synchronization log allowed me to quickly review if updates had been successfully downloaded to the SUS server.

Example 4: Text from SUS Approved log (edited)

Approved List Modified- November 06, 2002 3:00:10 AM Successful

Approved By: System (scheduled synchronization)

List of Approved Updates:

Windows XP Application Compatibility Update, April 2002

Q321599: Security Update

Q319733: Internet Information Services Security Roll-up Package

Security Update, March 4, 2002

EDITED to conserve space

List of Unapproved Updates:

Microsoft .NET Framework Service Pack 2, Japanese Version

Microsoft .NET Framework Service Pack 2, Japanese Version (SDK Applied)

Microsoft .NET Framework Service Pack 2, German Version

EDITED to conserve space

New Updates:

Q320174: Critical Update

Uniscribe (USP10 - CIF Only)

Security Update, May 30, 2001

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Example 5: Text from SUS Synchronization log (edited)

```
Automatic Sync Started- November 06, 2002 3:00:00 AM
    Software Update Services is up to date. No changes were required during synchronization.
Sync Finished- November 06, 2002 3:00:11 AM

Manual Sync Started- November 05, 2002 11:12:51 PM
    Software Update Services is up to date. No changes were required during synchronization.
Sync Finished- November 05, 2002 11:13:04 PM

Manual Sync Started- November 05, 2002 10:09:52 AM Successful
Updates Added:
    Security Update, April 2, 2001 - crlupd_6100C402FF3BC268CC77482180CBACECFC26A971.exe
Updates Removed:
    None
Reissued Update(s):
    None
Sync Finished- November 05, 2002 10:10:10 AM
```

The IIS log (Example 6) is complicated but can be helpful when troubleshooting connection or website response problems. The Deployment Guide is necessary to decipher the some of the SUS return code. [vii] (pgs. 77-84) Similarly, you will need an understanding of IIS to properly modify and decipher the IIS in this log.

Example 6: IIS log showing SUS activity (edited)

```
#Software: Microsoft Internet Information Services 5.0
#Version: 1.0
#Date: 2002-10-29 12:01:27
#Fields: date time c-ip cs-username s-sitename s-computername s-ip s-port cs-method cs-uri-stem cs-uri-query sc-status
sc-bytes cs-bytes cs(User-Agent)
2002-10-29 12:01:27 192.168.client - W3SVC1 USAYESSVR01 192.168.SUS 80 HEAD /iuident.cab 0210291155 200
241 135 Industry+Update+Control
2002-10-29 12:01:27 192.168.client - W3SVC1 USAYESSVR01 192.168.SUS 80 GET
/selfupdate/AU/x86/XP/en/wuaucomp.cab 2002-10-29 12:01:27 192.168.client - W3SVC1 USAYESSVR01 192.168.SUS
80 GET /wutrack.bin U=1d0c14c9c6f4cc46a5f9e449fb77c2fb&C=iu&A=n&I=&D=&P=5.1.a28.2.100.1.0&L=en-
US&S=s&E=00000000&M=&X=0210291155404322 00 241 237 Industry+Update+Control
```

NOTE: You can use IIS to modify the "fields" tracked in this log. The highlighted section shows some of the SUS return code.

You can use the Windows Update log (%systemroot%\Windows Update.log) and the System Event log to monitor the Automatic Update activity on the client machine. In one case, I had incorrectly configured the IP address of the SUS server for a particular client workstation; reviewing the Windows Update log identified the problem immediately.

In summary, the Microsoft Software Update Services requires a properly configured client and server in order to successfully deliver the updates. Client

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

configuration settings may be applied manually or delivered using Group Policies. The SUS server requires an Internet connection, for downloading updates, and IIS 5.0 or higher to handle internal clients requesting software updates. The SUS server settings are managed using web pages. Now that I felt like I understood the mechanics of the SUS solution, I created a test network (See Appendix B for diagram and description) to address the some concerns.

Resolving concern with the Software Update Service prior to implementation

1. **How does the SUS server communicate with AU clients and the Microsoft Windows Update website?**

In order to evaluate the security risks associated with the communication processes, I had to analyze the communication stream. I used the Ethereal Network Analyzer 0.9.7 to *sniff* the packets exchanged between the client and the SUS server during an update session. Notice how the process occurred via TCP and HTTP:80 requests. (See Table 1 for edited output) The SYN, SYN ACK, ACK pattern signals the beginning of the three-way handshake, followed by the request for the iuident.cab file, and finally the FIN, FIN ACK, pattern would conclude that initial transaction. I monitored the entire process and did not find that any other ports except TCP/HTTP:80 were used during the transaction. Furthermore, I noticed that the entire communication process was initiated by the AU client, the SUS server merely replied to requests. This information was vital in order to create a 'hole' in my firewall that would allow SUS access to the Internet without exposing the IIS service to Internet based attacks.

Table 1: Packets captured from Client / SUS Server exchange (edited)

No.	Time	Source	Destination	Protocol	Info
14	236.2763	192.168.client	192.168.server	TCP	1062 > http [SYN] Seq=218297017 Ack=0 Win=64240 Len=0
15	236.2768	192.168.server	192.168.client	TCP	http > 1062 [SYN, ACK] Seq=2507970218 Ack=218297018 Win=17520 Len=0
16	236.2769	192.168.client	192.168.server	TCP	1062 > http [ACK] Seq=218297018 Ack=2507970219 Win=64240 Len=0
17	236.2771	192.168.client	192.168.server	HTTP	GET /iuident.cab?0211060249 HTTP/1.1
18	236.2784	192.168.server	192.168.client	HTTP	HTTP/1.1 200 OK
19	236.2784	192.168.server	192.168.client	HTTP	Continuation
27	236.4132	192.168.client	192.168.server	HTTP	HEAD /selfupdate/AU/x86/XP/en/wuaucomp.cab?0211060249 HTTP/1.1
28	236.4172	192.168.server	192.168.client	HTTP	HTTP/1.1 200 OK
29	236.4173	192.168.client	192.168.server	TCP	1062 > http [FIN, ACK] Seq=218297310 Ack=2507978592 Win=64001 Len=0
30	236.4177	192.168.server	192.168.client	TCP	http > 1062 [ACK] Seq=2507978592 Ack=218297311 Win=17228 Len=0
31	236.418	192.168.server	192.168.client	TCP	http > 1062 [FIN, ACK] Seq=2507978592 Ack=218297311 Win=17228 Len=0

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

I used a similar process to monitor the packets exchanged between the SUS server and the Windows Update website. Again, the exchange occurred via TCP and HTTP:80 requests and at no time did the Microsoft Windows Update website initiate the connection. (See Table 2) Similarly, I used my Symantec Enterprise firewall log to monitor the transactions between the SUS server and Microsoft. I observed the firewall NAT process of masking the internal SUS server address before passing the packet to the Microsoft Windows Update website. (I had always wondered how that worked. The SANS Institute GSEC seminar taught me how firewalls use TCP port numbers to *remember* which internal client requested which external resource) The logs and output obtained from the sniffing utility, combined with a fundamental knowledge of TCP/IP transmission, allowed me to understand the mechanics of how the server and client communicate.

Table 2: Packets captured from SUS Server / Microsoft Windows Update website exchange (edited)

No.	Time	Source	Destination	Protocol	Info
954	5162.284	192.168.server	192.168.DNSserver	DNS	Standard query A www.msus.windowsupdate.com
955	5162.403	192.168.DNSserver	192.168.server	DNS	Standard query response A xxx.xxx.131.197 A xxx.xxx.131.229 A xxx.xxx.131.197 A xxx.xxx.131.229
956	5162.419	192.168.server	xxx.xxx.131.197	TCP	1061 > http [SYN] Seq=3675693476 Ack=0 Win=16384 Len=0
959	5162.419	xxx.xxx.131.197	192.168.server	TCP	http > 1061 [SYN, ACK] Seq=2434932104 Ack=3675693477 Win=17520 Len=0
960	5162.419	192.168.server	xxx.xxx.131.197	TCP	1061 > http [ACK] Seq=3675693477 Ack=2434932105 Win=17520 Len=0
961	5162.42	192.168.server	xxx.xxx.131.197	HTTP	GET /msus/v1/aucatalog.cab HTTP/1.1
962	5162.594	xxx.xxx.131.197	192.168.server	HTTP	HTTP/1.1 200 OK
963	5162.595	xxx.xxx.131.197	192.168.server	HTTP	Continuation
964	5162.595	192.168.server	xxx.xxx.131.197	TCP	1061 > http [ACK] Seq=3675693587 Ack=2434933828 Win=17520 Len=0
965	5162.595	xxx.xxx.131.197	192.168.server	HTTP	Continuation
966	5162.596	xxx.xxx.131.197	192.168.server	HTTP	Continuation
967	5162.596	192.168.server	xxx.xxx.131.197	TCP	1061 > http [ACK] Seq=3675693587 Ack=2434936485 Win=17520 Len=0
986	5171.295	192.168.server	xxx.xxx.131.197	TCP	1062 > http [SYN] Seq=3677932961 Ack=0 Win=16384 Len=0
987	5171.295	xxx.xxx.131.197	192.168.server	TCP	http > 1062 [SYN, ACK] Seq=4041570240 Ack=3677932962 Win=17520 Len=0
988	5171.296	192.168.server	xxx.xxx.131.197	TCP	1062 > http [ACK] Seq=3677932962 Ack=4041570241 Win=17520 Len=0
989	5171.296	192.168.server	xxx.xxx.131.197	HTTP	HEAD /msus/v1/aurtf.cab HTTP/1.1
990	5171.385	xxx.xxx.131.197	192.168.server	HTTP	HTTP/1.1 200 OK
991	5171.385	xxx.xxx.131.197	192.168.server	TCP	http > 1062 [FIN, ACK] Seq=4041570504 Ack=3677933069 Win=17413 Len=0
992	5171.385	192.168.server	xxx.xxx.131.197	TCP	1062 > http [ACK] Seq=3677933069 Ack=4041570505 Win=17257 Len=0
993	5171.385	192.168.server	xxx.xxx.131.197	TCP	1062 > http [FIN, ACK] Seq=3677933069 Ack=4041570505 Win=17257 Len=0

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Example 7: Symantec Enterprise Firewall hiding the SUS server IP address

```
Statistics: duration=10.45 id=a7gUj sent=110 rcvd=2920
srcif=insideNICMAC src=insidehost/1096
svsrc=firewallOutside/1360 dstif=firewallMAC
dst=xxx.xxx.131.197/80 op=GET
arg=http://www.msus.windowsupdate.com/msus/v1/aucatalog.cab
result="200 OK"

NOTE:
srcif = inside interface
src = inside host IP address
svsrc = service source IP address (outside IP address)
dstif = outside interface
```

2. Would the required Internet connection for SUS pose a risk to my internal network?

The SUS server requires an Internet connection in order synchronize and download updates from the Microsoft Windows Update website. I analyzed the connection process and determined that it was fully initiated by the SUS server. Therefore, I would **not** have to configure the firewall to 'listen' for requests from the Internet. This fact minimized the risk associated with outside attacks commonly associated with WORMS and script kiddies. I further 'restricted' the outbound connection by limiting Internet access from the SUS server, to the microsoft.com, windowsupdate.com and windows.com domains. These domains are used by the SUS server as source domains for the Internet update servers. I found these values in the IUServerURL section of the iuident.txt file. (See Example 8) This file is one of the files contained in the iuident.cab file normally exchanged during a SUS/Client transaction. The internet connection used by SUS does not present a significant risk to my *real world* network because it would require compromising, or circumventing, the firewall before an Internet based attack could hit the SUS server. Furthermore, if the server were compromised, the limited Internet access would reduce the potential for propagating code outside our network.

Example 8: List of Internet Update servers found in iuident.txt file

```
IUServerURLs]
ServerCount=4
Server1="http://v4.windowsupdate.microsoft.com/"
Server2="http://www.v4.windowsupdate.microsoft.com/"
Server3="http://www.v4.windowsupdate.com/"
Server4="http://v4.windowsupdate.com/"
```

3. Where would I locate my server on my LAN?

Our *real world* network is very small (See Appendix A for diagram and description) and does not currently support a DMZ or service network. The network is protected by a firewall that blocks all Internet traffic originating from

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

outbound sources. This configuration limits the attack vectors to the internal network and local console attacks. While the threat of viruses, malware, and inside hack attempts are real, the costs associated with constructing and maintaining a DMZ would not justify the cost of protecting the server. The SUS server will perform a single role and is not considered mission critical. If compromised, we could completely disable the server without affecting the daily business operations. Furthermore, the SUS server could easily be restored from backup and any patches downloaded after the initial moment of compromise would be considered 'tainted' and could be immediately resynchronized with the Microsoft Windows Update website. Therefore, the security risk associated with connecting this server directly to our LAN was acceptable.

4. Could IIS vulnerabilities be exploited by malformed URL requests?

I was concerned that someone could gain access to the system using a modified http request regardless of the fact that the SUS setup ran the IIS Lockdown tool. I inspected the file used to lock down the system (urlscan_static.ini) and was pleased to notice that the AllowVerbs section was limited to HEAD and GET and would severely limit the possibility of using the HTTP method to compromise the system. Furthermore, the DenyExtentions section would parse the URL string for common exploit strings like: root.exe, % characters and the ..used for traversing. Regardless, I searched my firewall log to and collected some of the more common attempts to compromise webservers. (See Example 9)

Example 9: URL strings used to attack SUS server

```
http://www.worm.com/default.ida?NNNN  <string edited to total 224 N's>
%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%u7801%u9090%u6858%ucbd3%
u7801%u9090%u9090%u8190%u00c3%u0003%u8b00%u531b%u53ff%u0078%u0000%u00=a
http://www/scripts/..%255c../winnt/system32/cmd.exe?/c+dir
http://www/scripts/..%%35%63../winnt/system32/cmd.exe?/c+dir
http://www/scripts/..%c0%af../winnt/system32/cmd.exe?/c+dir
http://www/_mem_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
http://www/_vti_bin/..%255c../..%255c../..%255c../winnt/system32/cmd.exe?/c+dir
http://www/d/winnt/system32/cmd.exe?/c+dir
```

I used the Netcat tool to send each of these strings to the SUS server. Fortunately, each request was denied by the URLScan filter as evidenced by the IIS log. (See Example 10)

Example 10: IIS log file showing rejected requests containing malformed URL requests (edited)

```
2002-11-07 14:25:41 10.72.80.76 - W3SVC1 USAYESSVR01 10.72.80.184 80 GET /<Rejected-By-UrlScan> ~/http/1.0 404
4184 14 -
2002-11-07 14:43:47 10.72.80.76 - W3SVC1 USAYESSVR01 10.72.80.184 80 GET /<Rejected-By-UrlScan>
~http://www/scripts/..%c0%2f../winnt/system32/cmd.exe 404 4184 64 -
2002-11-07 15:38:01 10.72.80.76 - W3SVC1 USAYESSVR01 10.72.80.184 80 GET /<Rejected-By-UrlScan> ~/default.ida 404
4184 380 -
2002-11-07 15:42:29 10.72.80.76 - W3SVC1 USAYESSVR01 10.72.80.184 80 GET /<Rejected-By-UrlScan>
~http://www/scripts/dd/winnt/system32/root.exe 404 4184 49 -
2002-11-07 15:46:28 10.72.80.76 - W3SVC1 USAYESSVR01 10.72.80.184 80 GET /<Rejected-By-UrlScan> ~//www/ 404 4184
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

5. **Could the SUS serve be used to distribute malware disguised as an update?**

YES! (I discovered this after I had written this paper AND implemented the solution on a small network. I've since reconfigured the SUS server to redirect my AU Clients to the Microsoft Windows Update website.)

This was my largest concern because of two reasons: First, the updates are stored on the SUS server as uncompressed and fully functional executable files, and second, would SUS distribute an infected file? In order to test the concept, I accessed server using administrative access. (Obviously, the server would be considered compromised at this point, but I wanted to see if the server could be used to quietly distribute rouge executables disguised as operating system updates). The results were discouraging. (Table 3 summarizes my test results)

Table 3: Results from tampering with files in the .cab folder

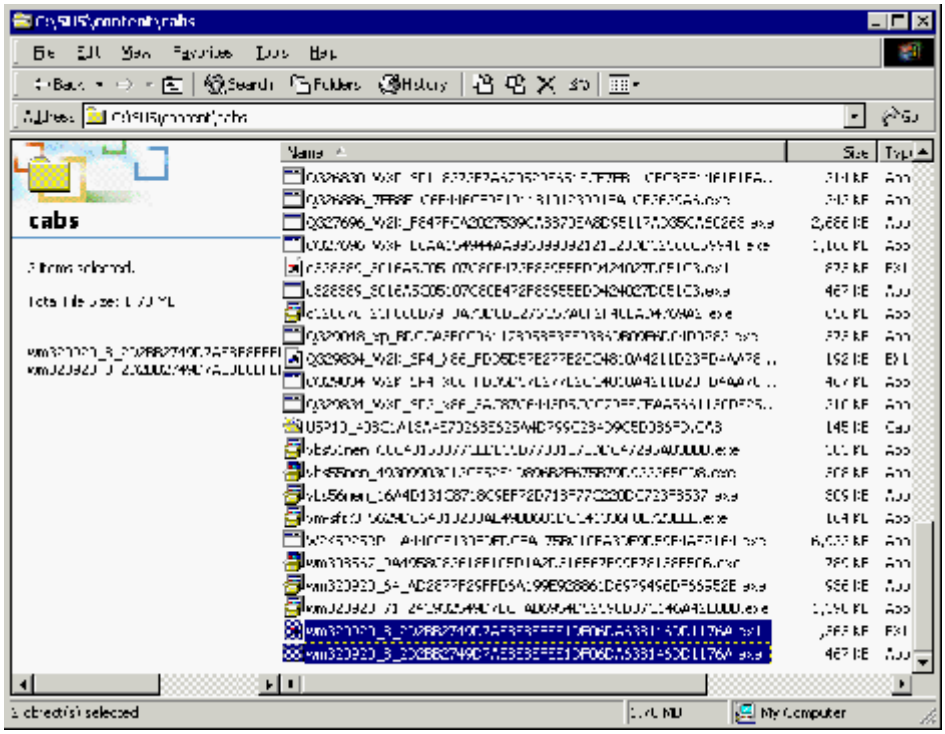
Test Performed	Threat Assessment		
	Server	Client	Overall
Delete a file in the .cabs folder	LOW: SUS resynchronized the file on next cycle.	LOW: Client could not install patch until it was resynchronized by SUS.	LOW: Does not actively harm Server or Client. Server will automatically recover
Rename existing file with different name	MED/LOW: SUS resynchronized the file on next cycle but left renamed file in the folder	LOW: Client could not install patch until it was resynchronized by SUS.	MED/LOW: Does not actively harm client but ignored files in a folder could become malicious in the future.
Copied a un-related file to folder	MED/HIGH: SUS ignored the file.	MED/LOW: Client does not try to download the file, but file could become malicious in the future.	MEDIUM: Any file left stranded in the folder could become an active and malicious at a later date.
Replaced an existing file with a different file of the same name.	HIGH: SUS left the file there and did not replace during synchronization cycle. This could allow a hacker to place code disguised as patch	HIGH: Client executed the file as if it were a Microsoft update.	HIGH: The server can be compromised and used to quietly distribute rouge software disguised as a Microsoft update.

The results of the last two tests, copying an unrelated file to the update folder and replacing an existing file demonstrated a huge vulnerability. **If a hacker can place a file SUS system disguised as an official update, then that code will be deployed to every AU client that chooses to install that update**.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

I proved this by copying the hfnetchk.exe file to the \cabs folder and renaming it as a legitimate update. (See Example 11)

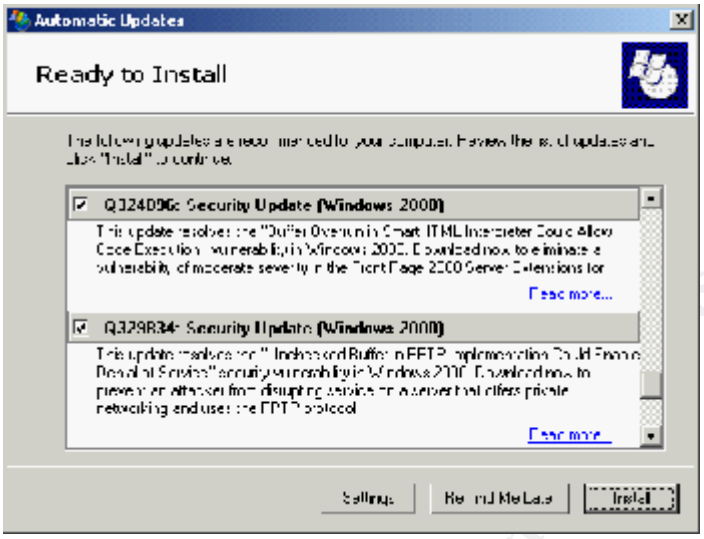
Example 11: Rouge file renamed as legitimate update



Then I synchronized SUS, hoping that SUS would overwrite the rogue file. SUS didn't recognize the file size, or date modified, discrepancy and left the rouge file untouched. (First red flag) Then I configured a test AU client to notify me before downloading and installing updates. I triggered the AU client to poll the SUS and was informed that an update was available. SUS continues to announce the rouge file as a viable update. (Second red flag) I allowed the AU client to download the update and was presented with the details of the update, it looked legitimate. (See Example 12)

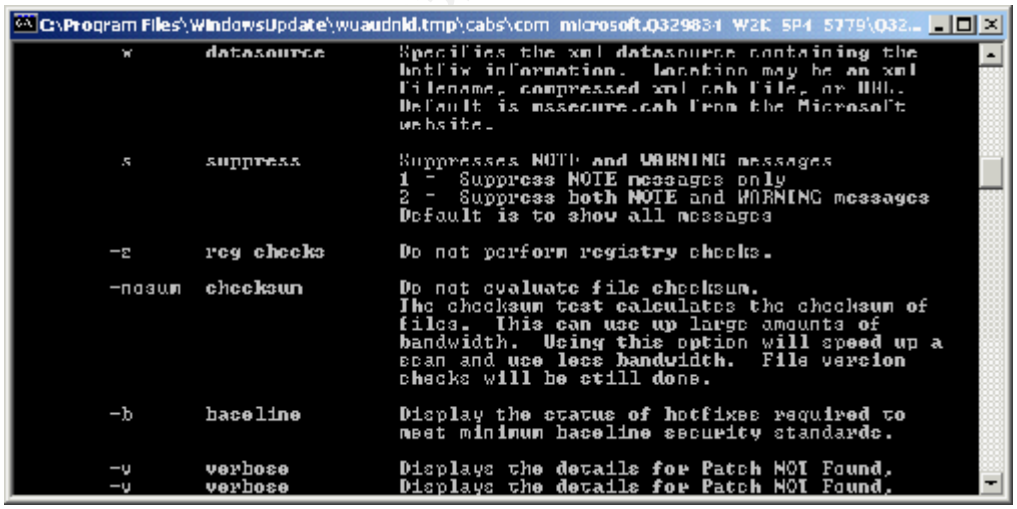
© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Example 12: SUS announces that updates are available, including my rogue file.



Then I clicked the install button and crossed my fingers. A command window popped up and ran the rouge file (hfnetchk.exe). I only noticed this because if you don't use the proper syntax, it will display a lengthy explanation of the available switches used for that command. I was able to capture the screen before it disappeared, notice how the title bar reflects the file being executed. (See Example 13)

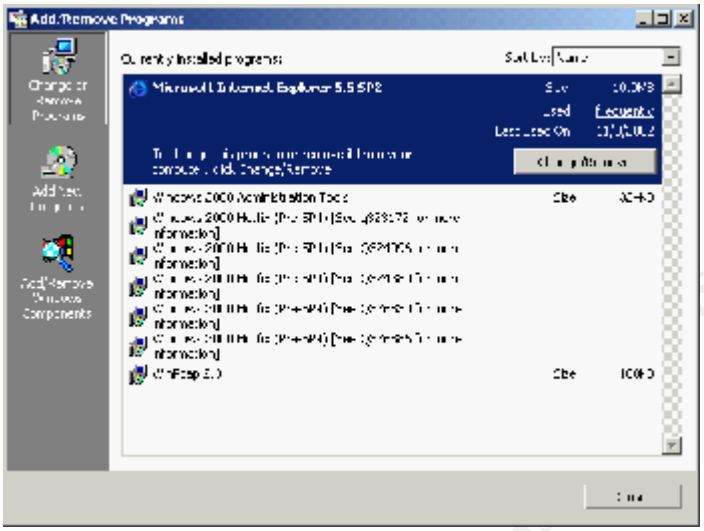
Example 13: Rouge file being executed as a Windows Update



The window closed and to my horror, I was presented with a screen telling me that updates had been successfully installed and would require a restart. After restarting I checked the Add/Remove programs and did NOT find the update listed. (See Example 14)

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Example 14: Rouge patch did not appear as an installed update



This was a very crude test; true code writers could create a package that would launch a legitimate update while quietly running their malicious code in the background. This behavior represents a large security risk in any environment. I have one suggestion that would reduce this threat. During the initial SUS setup (See Appendix D, screen #4) you can choose to redirect your clients to the Microsoft Windows Update website. This allows the administrator to control which updates may be applied, but does not keep the content on the local server. When the AU client polls the internal SUS server it will be presented with a list of approved updates. The download process will contact the Microsoft site. This allows for greater control of updates, but does require that the AU client has Internet access. This would reduce the risk of a hacker using the SUS server to distribute rogue software, however it will introduce the risks associated with allowing your clients access to the Internet. (A whole new set of security issues)

Rolling out Software Update Services

WARNING: Be sure that you are aware of the security risk associated with this product!

The Software Update Services solution has a minimum software level requirement that may exclude many small networks that do not have the financial resources necessary to bring all workstations up to Windows 2000 or better. Fortunately, my *real world* company already had Windows 2000 Professional and XP workstations.

Prepare the workstations for the Automatic Update client

The most time consuming step involved manually updating each workstation, I had to install Service Pack 3 and IE 5.5 on each workstation in order to meet the minimum software requirements for the Automatic Update service. This served as the baseline for all my machines. I used the

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

hfnetchk.exe utility to verify that each machine was at the baseline before deploying the AU client. The important thing to remember is that SUS will continue to maintain the updates until the next Service Pack is released.

Install SUS on a dedicated server

Once the workstations were ready, I concentrated on building the server. My *real world* company had an older server that was no longer in service because it had been replaced with a newer model. The old server, an HP NetServer E45, P6233 with 256MB RAM and a 4GB drive was perfect since SUS would require a dedicated server. I installed Windows 2000 with the minimum components necessary and limited the IIS components to **Common Components, Internet Information Services Snap-In** and the **World Wide Web Server** as suggested by the deployment guide. [vi] (pg. 65) Then, I downloaded and installed SUS server component. (Please refer to Appendix D for the SUS installation screen shots and additional comments.) During the setup, I choose the store the updates on the local server. (WARNING: A recently discovered security risk (11/11/2002) would suggest that you choose to redirect your AU clients to the Microsoft Windows Update website.) Next, I chose English, as the only language to download in order to conserve space in the hard disk. I chose to "manually" approve the updates rather than allowing SUS to automatically approve new updates. This was the best option for my *real world* network because I could review and test the updates before releasing them to the workstations. After finalizing the installation, I had to prepare the firewall to allow the SUS server to access the Internet.

Open an Internet connection in the firewall for the SUS server

My *real world* network uses a Symantec Enterprise firewall to protect the inside LAN. This firewall allows me apply rules to objects called "Network Entities". I created an entity called "SUS server" and allowed it limited access to the Internet. I restricted outbound HTTP connections originating from the SUS server, destine for the Internet, to the microsoft.com, windowsupdate.com and windows.com domains knowing that these were the only domains were used to download updates. This restriction will limit the ability to use my internal web server to attack other Internet web servers.

Block clients from using the Windows Update website

If the AU settings are configured by a Group Policy, then the local settings cannot be overwritten, however, this does not prevent the end user from clicking on the Start Menu shortcut or manually navigating to the Windows Update website to apply patches. I needed to prevent my end users from circumnavigating the SUS solution so I began by deleting the shortcut from the Start Menu; 'security by obscurity'. Then, I created a firewall group entity called "NoUpdates" and added the AU clients that group. Next I wrote a rule that forced all HTTP traffic from originating from the "NoUpdates" group destine for the

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

"Universe" to be passed through a URL pattern filter (http.urlpattern). Any URL string that matched one in the filter would be denied. I added the string **.*\v4.windowsupdate.microsoft.com*** to prevent users from using their Internet connection to manually add unapproved updates. This would allow the SUS server to access the Windows Update website and still allow my *real world* hosts access to the Internet. (NOTE: Due to a recently discovered vulnerability, I had to temporarily remove this restriction and allow the host machines access to the Windows Update website; however a recently created security policy warns against doing that.)

Synchronize the SUS server

Once the Internet connection was ready, I began the process of synchronizing the server. This is where the SUS server contacts the Microsoft Windows Update website and downloads the appropriate language updates. First I had to connect to the server, which was easy since the installation placed a shortcut on the desktop. I could have accessed the SUS administrator website from any workstation on the LAN by navigating to http://*servername*/susadmin and entering the proper credentials. Once authenticated, I began the initial download which consumed approximately 150mb of disk space. (NOTE: Due to a recently discovered security risk, you may wish configure the SUS server to redirect your AU clients to the Microsoft Windows Update website.)

Approve the updates

Once the synchronizing was complete, I had to manually select each update and check the "Approve" box. This is a daunting task at first because the initial download 'fetches' over 70 updates. One annoyance involved the many foreign language .NET patches that were downloaded despite the fact that I didn't ask any foreign language updates. The process of manually approving the updates will prevent my end users from applying updates before they can be tested for compatibility. (Note: This process remains valid if you choose to store the updates on a Microsoft server. Your AU clients will download approved updates directly from Microsoft.)

Apply the Automatic Update settings to your clients

I used the Group Policy editor to configure the Automatic Update settings for client computers in my Active Directory tree. Using the "AD Users and Computer" console, I created a new Group Policy called "Windows Update" that would automatically apply AU settings to all computers. This process involved using the Group Policy editor to create a new Administrative Template. Use the WUAU.adm template found in the %windir%/inf folder. (See Example 2 for screen shots) I configured service to automatically check the SUS server daily at 03:00 and automatically apply the updates. This insured that the updates would be applied without relying upon end user intervention. (I instructed the users to log off and leave their PC's powered on overnight.)

Wait and verify that the service is working

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Once everything was configured, I verified that the process was working. I checked the IIS logs and noted that each client had attempted to contact the SUS website, but this wouldn't tell me if the patches were successfully applied. I used the hfnetchk.exe utility to remotely check the workstations (See Appendix C for example output) Administrators can check the Windows Update log or System Event log too. I developed a process of verifying the workstations weekly by using the hfnetchk.exe utility and a host file to check each workstation from a single command. I piped the output to a text file and print it out every Monday morning. This enabled me to insure that each workstation was continuing to remain at the current update level.

Conclusion

My initial objective was to find a solution that would allow me to deliver and install the latest patches to every workstation and require minimal end user intervention. The solution had to be inexpensive and require minimal administrative resources. The *real world* client had an extra server that was not being used, so the cost of dedicated hardware was not a concern. Most small companies might not be in a position to purchase new hardware for a solution like this, but if the machine can run W2K and IIS 5 or better, and is not used as a domain controller, then it can be used for SUS. The largest administrative task involved updating each workstation BEFORE I could install the Automatic Update service. (W2K SP2/IE 5.5+) That task alone significantly increased the security posture for each workstation. Once that was complete, the process of configuring the clients was handled using Group Policies. The best part of the solution was that I (an administrator) could control which patches would be available for the workstations. This allowed me to review and determine which patches would be safe to deploy in our network which directly addressed the problem of end users applying patches unnecessarily. The restricted Internet connection coupled with domain level policy settings prevented my end users from overriding the SUS solution. While the solution did address my initial problem, the SUS solution does have some noteworthy limitations. First, you cannot assign approved updates to a particular machine or user group; all approved updates are available to every AU client. Second, the SUS does not download or distribute Service Packs or other Microsoft product updates. Furthermore, an administrator cannot choose which updates to download, their automatically sent from Microsoft.

Despite the product limitations, if you are an administrator of a very small network, and are concerned that operating system security and software updates are not being diligently applied to your workstations or file servers, then consider Microsoft's Software Update Services. It's a useful tool that will assist you in delivering critical software updates to hosts without a requiring a considerable amount of financial or administrative resources. (WARNING: A recently discovered vulnerability may make this product unsafe for your environment unless properly configured. Please see Concern #5 for details)

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

AUTHOR'S NOTE. Patchlink Update 4.0 (http://www.patchlink.com) has a free 10 user license available. You must qualify for the license and are required to pay for continued maintenance after a year. See their website for details.

© SANS Institute 2003, Author retains full rights

References and Resources

Cited Internet URL References:

(i) Pawlak, Peter. "Software Update Service to Ease Patch Distribution." Apr. 22, 2002. URL: http://www.directionsonmicrosoft.com/sample/DOMIS/update/2002/05may/0502sustep.htm (9/30/2002).

(ii) "Microsoft Network Security Hotfix Checker (Hfnetchk.exe) Tool Is Available." Q303215, Oct. 10, 2002. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;Q303215 (10/15/2002).

(iii) The SANS Institute. "Mistakes People Make that Lead to Security Breaches". October 23, 2001. URL: http://www.sans.org/mistakes.htm (10/1/2002)

(iv) Microsoft Corporation. "Background Intelligent Transfer Service". August 2002. URL: http://msdn.microsoft.com/library/en-us/bits/drzportal_06xx.asp (September 30, 2002) (NOTE: When verifying this link on November 9, 2002, I received a 404 error. However, this document contains the both the original link I used, and helpful information about BITS. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwxp/html/WinXP_BITS.asp)

(v) Hill, Brett. "IIS Admins: Prepare BEFORE installing Software Updates Services" URL: http://www.iisanswers.com/articles/sus.htm (Oct. 20 2002)

(vi) eEye Digital Security. "Code Red" Worm" July 17, 2001. URL: http://wwww.eeye.com/html/Research/Advisories/AL20010717.html (September 3, 2002)

(vii) Microsoft Corporation. "Deploying Microsoft Software Update Services". August 2002. URL: http://www.microsoft.com/windows2000/windowsupdate/sus/susdeployment.asp (September 30, 2002)

Culp, Scott. "Where to Find Microsoft Security Patches". February 2001. URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/columns/security/essays/secpatch.asp

Microsoft Corporation. "How to Force Automatic Updates 2.2 to Perform a Detection Cycle". August 8, 2002. URL: http://support.microsoft.com/default.aspx?scid=kb;en-us;326693 (November 1, 2002)

Microsoft Corporation. "Software Update Services Overview White Paper". June 20, 2002. URL: http://www.microsoft.com/windows2000/windowsupdate/sus/susoverview.asp (August 28, 2002)

Fire Tower Inc. "How do I prevent CodeRed and Nimda exploit attempts from making it to my web server". September 20, 2001. URL: http://www.firetower.com/faqs/general/httpurlpattern.html (October 12, 2002)

SANS books

(ix) Shawgo, Jeff, Editor. "Windows 2000 Security: Step by Step" Version 1.5. The SANS Institute. July 1, 2001

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Software Used

SUS server
46.17 MB file
3 hr 44 min @ 28.8 Kbps
http://www.microsoft.com/Windows2000/downloads/recommended/susserver/download.asp

SUS client agent
1000 KB file
5 min @ 28.8 Kbps
http://www.microsoft.com/Windows2000/downloads/recommended/susclient/download.asp

Microsoft Hot Fix Checking utility
Version 3.3
252kb
http://www.microsoft.com/downloads/release.asp?releaseid=31154

Ethereal Network Analyzer
Version 0.9.7
Win32 Binary Version
http://www.ethereal.com/distribution/win32/

Netcat for NT/2000/XP
Version 02.08.98
96kb
http://www.atstake.com/research/tools/

© SANS Institute 2003, Author retains full rights

APPENDIX A

"Real world" network diagram and description

This is a homogeneous network consisting of one Windows 2000 file server with Active Directory, and 10 Windows 2000 Professional hosts. It is secured behind a Symantec Enterprise firewall.

Figure 1: Network description before installing SUS sever

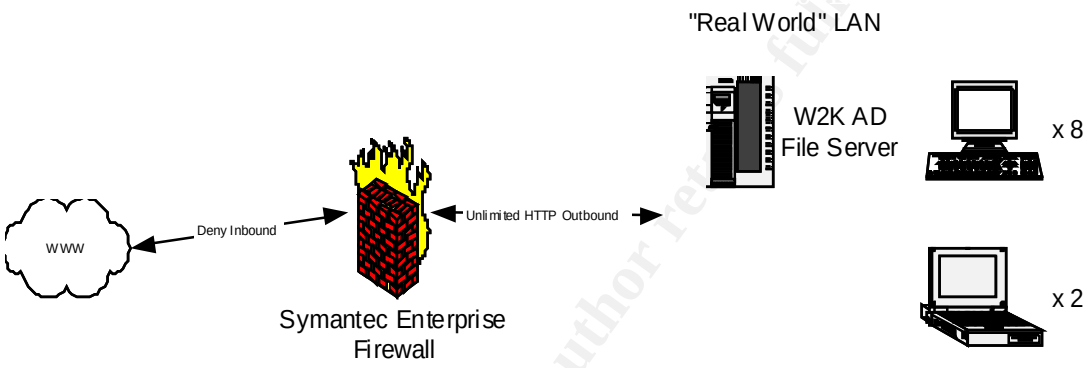
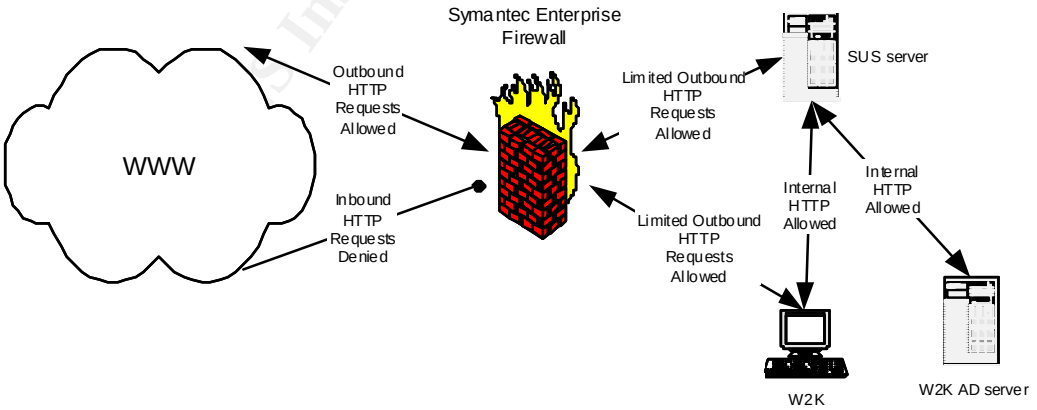


Figure 2: Network description AFTER installing SUS sever

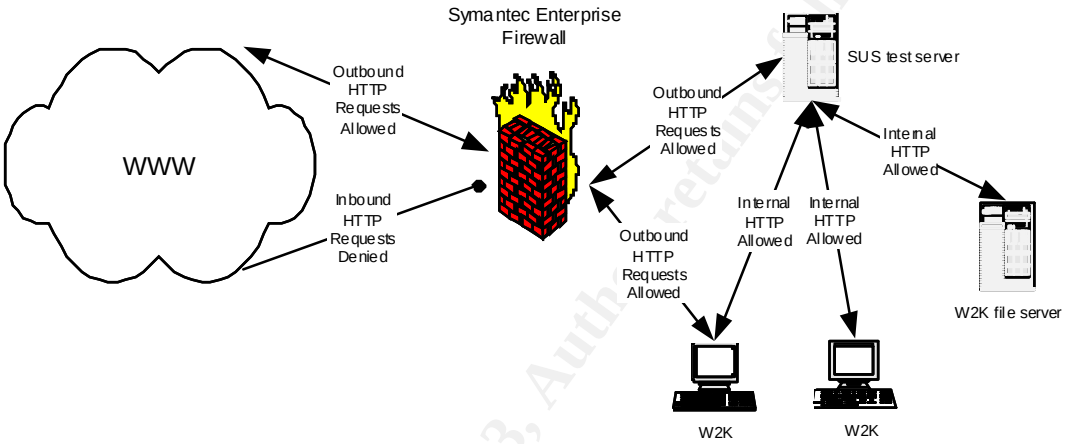


© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

APPENDIX B

Test network diagram and description

The test network had 2 file servers and 3 workstations behind a Symantec Enterprise Firewall. The firewall is configured to allow the SUS file outbound TCP HTTP:80 requests to the Internet. The other file server and workstations cannot access the Internet. The firewall will not allow any incoming HTTP requests to any internal host. I chose this configuration to closely match the existing production network.



W2K Servers:

Please review the security SANS "Securing Windows 2000"[ix] or similar guideline, before deploying ANY W2K server in a production environment.

I decided to use a default "vanilla" installation of Windows 2000 server on both servers. I installed Service Pack 3. On the server hosting SUS, I installed IIS 5.0 using the minimum SUS requirements: Common files, Microsoft Management Console snap-in, World Wide Web services. I did not perform any of the additional OS or IIS hardening techniques. I felt that this environment would more closely resemble the configurations found in "real-world" small networks where less experienced administrators are responsible for implementing a solution like this.

Workstations:

I began by performing a default "vanilla" installation for each workstation. SUS requires an updated version of the Automatic Update utility already found on Windows 2000; therefore I upgraded the browsers to Internet Explorer 5.5 and applied Service Pack 3.

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

APPENDIX C

Using Microsoft's patch checking utility to verify results

You can use Microsoft's Hot Fix Patch checking utility to compare the actual service pack, patch and hot fix state of a machine against Microsoft's recommended minimum requirements. This command line utility will download an XML document to the local machine and use it to determine if patches are missing from a particular machine. I often use this utility to verify if a machine has the appropriate patches applied. Use the –h *hostname* switch to check remote hosts.

Listed below are two examples taken from 2 systems with identical hardware and software on my *real world* network before the SUS solution was implemented. The output should be identical for these Windows 2000 machines. Note the differences before the SUS solution was applied compared to the results to after Service Pack 3 and SUS was implemented. Security and post IE5.5 updates were equally applied to each workstation.

HFNETCHK results BEFORE the SUS solution was implemented

```
Windows 2000 host #1
------------------------
HOSTNAME (192.168.X.X)
------------------------

        * WINDOWS 2000 SP2

        Warning
        The latest service pack for this product is not installed.
        Currently SP3 is installed.
        Note              MS01-022        Q296441
        Patch NOT Found   MS02-001        Q311401
        Patch NOT Found   MS02-006        Q314147
        Patch NOT Found   MS02-013        Q300845
        Patch NOT Found   MS02-014        Q313829
        Patch NOT Found   MS02-017        Q311967
        Patch NOT Found   MS02-024        Q320206
        Patch NOT Found   MS02-029        Q318138
        Patch NOT Found   MS02-042        Q326886
        Patch NOT Found   MS02-045        Q326830
        Patch NOT Found   MS02-048        Q323172
        Patch NOT Found   MS02-050        Q328145
        Note              MS02-053        Q324096
        Patch NOT Found   MS02-055        Q323255

        * INTERNET EXPLORER 5.5 SP1

        Warning
        The latest service pack for this product is not installed.
        Currently SP1 is installed.  The latest service pack is
        Internet
        Explorer 5.5 SP2.
        Patch NOT Found   MS02-009        Q318089
        Patch NOT Found   MS02-047        Q323759
```

```
Windows 2000 host #2
------------------------
HOSTNAME (192.168.X.X)
------------------------

        * WINDOWS 2000 SP2

        Warning
        The latest service pack for this product is not installed.
        Currently SP2 is installed.  The latest service pack is SP3.
        Note              MS01-022        Q296441
        Patch NOT Found   MS02-014        Q313829
        Patch NOT Found   MS02-017        Q311967
        Patch NOT Found   MS02-024        Q320206
        Patch NOT Found   MS02-029        Q318138
        Patch NOT Found   MS02-042        Q326886
        Patch NOT Found   MS02-045        Q326830
        Patch NOT Found   MS02-048        Q323172
        Patch NOT Found   MS02-050        Q328145
        Note              MS02-053        Q324096
        Patch NOT Found   MS02-055        Q323255

        * INTERNET EXPLORER 5.5 SP2

        Patch NOT Found   MS02-047        Q323759
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

HFNETCHK results AFTER the SUS solution was implemented

```
Windows 2000 host #1
--------------------------
HOSTNAME (192.168.X.X)
--------------------------

        * WINDOWS 2000 SP3

        Note              MS01-022        Q296441
        Patch NOT Found   MS02-050        Q328145
        Note              MS02-053        Q324096
        Note              MS02-064        Q327522

        * INTERNET EXPLORER 5.5 SP2

        Information
        All necessary hotfixes have been applied.
```

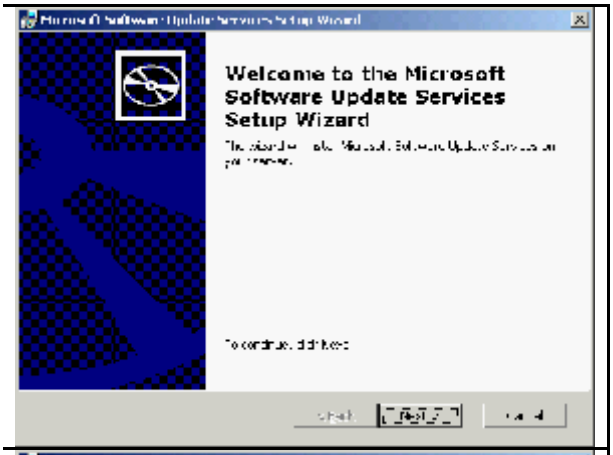
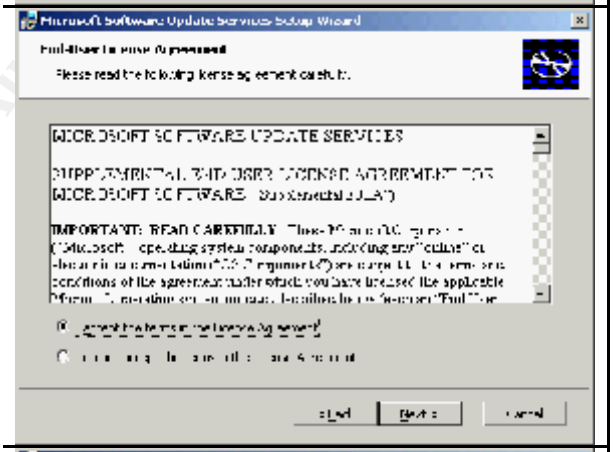
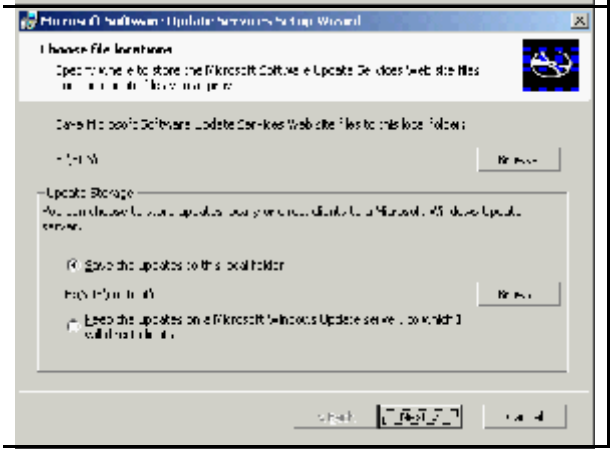
```
Windows 2000 host #2
----- ------------------
   HOSTNAME (192.168.X.X)
--------------------------

* WINDOWS 2000 SP3

        Note              MS01-022        Q296441
        Patch NOT Found   MS02-050        Q328145
        Note              MS02-053        Q324096
        Note              MS02-064        Q327522

        * INTERNET EXPLORER 5.5 SP2

        Information
        All necessary hotfixes have been applied.
```

© SANS Institute 2003, Author retains full rights.

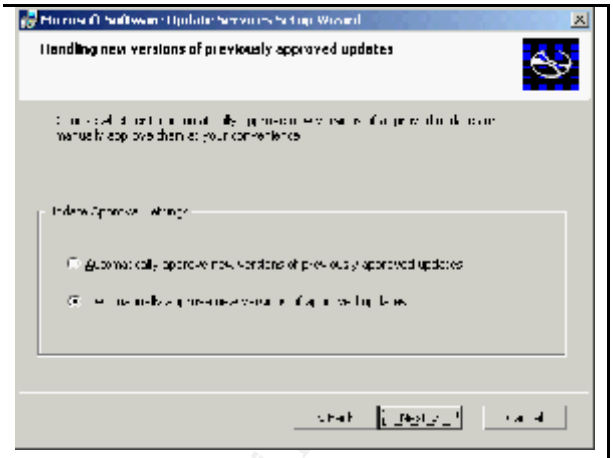
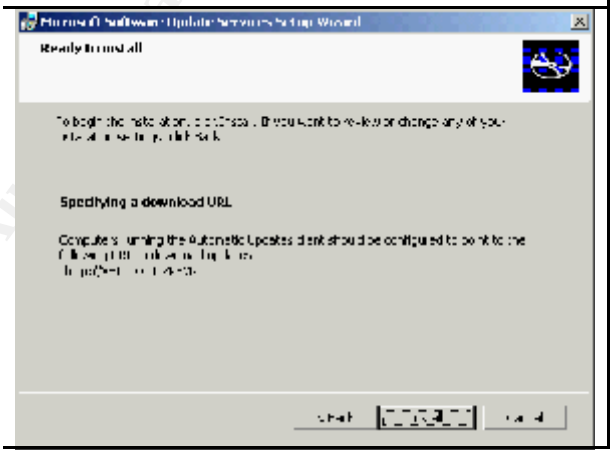
APPENDIX D:

Installing Software Update Service

The following are a list of screenshots presented if when choosing "Custom" rather than the "Typical", installation.

Screen 1 Begin wizard	
Screen 2 You must accept the EULA before proceeding	
Screen 3 Not shown, choose the CUSTOM setup. **Screen 4 (CAUTION)** Choose where you want the updates to be stored. Updates can be stored on the machine OR you can redirect your hosts to the Microsoft Windows Update server. (Recommended)	

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Screen 5 Not shown. You must choose a language. (Choosing additional languages will increase the amount of disk space and download time required to synchronize your server. **Screen 6** After choosing a language, You can choose to manually or automatically approve the updates. If you choose "Manual" then the server would download the updates but not make them available to inside hosts until they were approved. This is perfect for administrators wishing to test the update before deployment.	
Screen 7 This value will be reported to the hosts as the "root" URL. Make sure your hosts can resolve this name; otherwise you may use the IP address. The install will begin copying files and present you with the final "Finish" screen.	

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

Appendix E

Scripts and batch files used during case study

NFTS permissions batch file

This batch file uses CACLS and is useful for determining NTFS permissions on a folder and/or nested folders. Copy the script into a batch file called dirperm.bat

Syntax:
dirperm “path of folder” “name of file for output”

Example:
Dirperm.bat “c:\sus\content\cabs” “c:\cabs.txt”

Batch file used to view NTFS permissions

```
@echo off
if exist %2 del /q %2
if /i "%3"=="/s" goto sub
CACLS %1 >> %2
CACLS %1\*.* >>%2
exit
:sub
if exist %TEMP%\DirPerms.srt del /q %TEMP%\DirPerms.srt
if exist %TEMP%\DirPerms.log del /q %TEMP%\DirPerms.log
CACLS %1 >> %2
for /f "Tokens=*" %%i in ('dir %1\*.* /B /AD /ON /S') do echo
%%i>>%TEMP%\DirPerms.srt
sort <%TEMP%\Dirperms.srt >%TEMP%\dirperms.log
for /f "Tokens=*" %%i in ('type %TEMP%\dirperms.log') do CACLS "%%i ">>%2
del /q %TEMP%\dirperms.srt
del /q %TEMP%\dirperms.log
exit
```

Output from batch file

```
c:\sus\content\cabs USAYESSVR01\Web Applications:(OI)(CI)(DENY)(special access:)

                                DELETE
                                WRITE_DAC
                                WRITE_OWNER
                                FILE_WRITE_DATA
                                FILE_APPEND_DATA
                                FILE_WRITE_EA
                                FILE_DELETE_CHILD
                                FILE_WRITE_ATTRIBUTES

        USAYESSVR01\Web Anonymous Users:(OI)(CI)(DENY)(special access:)

                                  DELETE
                                  WRITE_DAC
                                  WRITE_OWNER
                                  FILE_WRITE_DATA
```

© SANS Institute 2003, As part of the Information Security Reading Room. Author retains full rights.

```
                                FILE_APPEND_DATA
                                FILE_WRITE_EA
                                FILE_DELETE_CHILD
                                FILE_WRITE_ATTRIBUTES

        Everyone:(OI)(CI)(special access:)

                READ_CONTROL
                SYNCHRONIZE
                FILE_GENERIC_READ
                FILE_READ_DATA
                FILE_READ_EA
                FILE_READ_ATTRIBUTES

        BUILTIN\Administrators:(OI)(CI)F
        NT AUTHORITY\SYSTEM:(OI)(CI)F
        Everyone:(OI)(CI)(special access:)

                READ_CONTROL
                SYNCHRONIZE
                FILE_GENERIC_READ
                FILE_READ_DATA
                FILE_READ_EA
                FILE_READ_ATTRIBUTES

        BUILTIN\Administrators:(OI)(CI)F
        NT AUTHORITY\SYSTEM:(OI)(CI)F
```

© SANS Institute 2003, Author retains full rights

Last Updated: May 20th, 2013

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Malaysia @ MCMC 2013	Cyberjaya, MY	Jun 03, 2013 - Jun 08, 2013	Live Event
SANS Pen Test Berlin 2013	Berlin, DE	Jun 03, 2013 - Jun 08, 2013	Live Event
Industial Control Systems Security Training - Houston	Houston, TXUS	Jun 10, 2013 - Jun 15, 2013	Live Event
Security Impact of IPv6 Summit 2013	Washington, DCUS	Jun 14, 2013 - Jun 16, 2013	Live Event
SANSFIRE 2013	Washington, DCUS	Jun 14, 2013 - Jun 22, 2013	Live Event
SANS Canberra 2013	Canberra, AU	Jul 01, 2013 - Jul 13, 2013	Live Event
Digital Forensics & Incident Response Summit 2013	Austin, TXUS	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS London Summer 2013	London, GB	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS Rocky Mountain 2013	Denver, COUS	Jul 14, 2013 - Jul 20, 2013	Live Event
SANS Mumbai 2013	Mumbai, IN	Jul 22, 2013 - Jul 27, 2013	Live Event
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	Washington, DCUS	Jul 22, 2013 - Jul 26, 2013	Live Event
SANS San Francisco 2013	San Francisco, CAUS	Jul 29, 2013 - Aug 03, 2013	Live Event
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	Bangalore, IN	Aug 05, 2013 - Aug 10, 2013	Live Event
SANS Boston 2013	Boston, MAUS	Aug 05, 2013 - Aug 10, 2013	Live Event
Critical Security Controls Summit	Washington, DCUS	Aug 12, 2013 - Aug 18, 2013	Live Event
Industrial Control Systems Security Training - DC	Washington, DCUS	Aug 12, 2013 - Aug 16, 2013	Live Event
SANS Thailand 2013	Bangkok, TH	Aug 19, 2013 - Aug 31, 2013	Live Event
SANS Virginia Beach 2013	Virginia Beach, VAUS	Aug 19, 2013 - Aug 30, 2013	Live Event
Mobile Device Security Summit 2013	OnlineCAUS	May 30, 2013 - Jun 06, 2013	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced