Securing the Symantec LiveUpdate Administrative Utility on Windows 2000

Cedric Albis

Securing the Symantec LiveUpdate Administrative Utility on Windows 2000

Cedric d'Albis
9th August 2002
GSEC assignment version 1.4, option 1

1 Executive Summary

This paper describes in detail the steps required to implement and harden a Symantec LiveUpdate server on a Microsoft Windows 2000 platform. Such a server can greatly reduce the bandwidth required by Norton Antivirus client machines across a campus or enterprise network. It can also prevent denial of service attacks against Norton Antivirus client machines and control the rollout of virus updates. Like any server connected to the Internet, it should be hardened (turned into a bastion host) in order to prevent malicious outsiders from accessing and exploiting it.

In addition to being a cookbook to build a LiveUpdate FTP server, this paper describes methods and concepts that can be used to secure any vendor application on the Windows 2000 platform. Furthermore, the methods described herein are all native to Windows 2000 (N.B. auditing tools excepted). Thus insuring that these methods can be used consistently in large environments at no additional costs.

2 Deployment Scenario

Norton Antivirus (NAV) is a popular anti-virus solution sold by Symantec. There are two parts to the software: a scanning engine that analyses files; and a database of virus "signatures". The scanning engine detects infected files by comparing files, or parts of files, to the signatures present in the database. The main implication from this architecture is that the effectiveness of the software is greatly dependent on the database. For NAV to remain effective against new forms of malware, these signatures have to be regularly updated.

To insure that NAV and other products (e.g. PCAnywhere, Norton Internet Security) are updated on a regular basis, Symantec offers the aptly named LiveUpdate agent. When this agent runs, it checks for the presence of various Symantec products on the machine, checks the product version, connects to a Symantec site (using http or ftp), downloads updates if there are updates available and, finally, installs the updates.

In a large environment, it is a waste of bandwidth to let LiveUpdate clients on hundreds, or thousands of machines contact the Symantec site and download the same files. Symantec has addressed this issue by providing the LiveUpdate Administration utility. When this utility runs, it does not check for locally installed applications, but downloads all the available updates. LiveUpdate clients can then connect to this server rather than to the Symantec site.

More importantly, the LiveUpdate Administration utility lets administrators control which updates are available to clients across the network, thereby allowing updates to be tested in a safe staging environment before being rolled out. Finally, the LiveUpdate Server protects clients against attacks based on spoofing the Symantec FTP or Web site.

3 The Application

The most common problem encountered when hardening a server is that at the end of the procedure, the server is secure but the application does not work. At that stage, you might as well take the additional precautions of cutting the cable(s) connecting the server to the network, casting it in a block of poured concrete and dropping it deep in the ocean.

Of course, this will not win you many friends in the user community. So, before changing any configurations on the server, it is crucial to understand how the application works: how it interacts with the OS, which services it depends on, and which protocols it uses.

Some would argue that the best source of information is (pun intended) the source code of the application; but source code is not always available and even when it is, it can be very difficult and time-consuming to analyse.

An even better way to insure that an application runs in a secure way on a hardened server is to engage and involve the developers, by giving them access to hardened builds and hardening documentation. This type of collaborative approach is seldom possible, but must be pursued whenever possible (i.e. working with in-house applications).

In the majority of cases on the Microsoft Windows 2000 (WIN2K) platform today, the approaches mentioned cannot be used because system administrators have to work with third party, proprietary, closed source software. For these cases, we will have to use methods that allow us to understand how the application works without access to the source code or to the developers. These are the methods described below.

3.1 Installing the application

3.1.1 Choosing the features to install

Modern applications are multifunctional "beasts", and Norton Antivirus Corporate edition is no exception. The Corporate Edition offers many management features as part of the Symantec System Center (a Microsoft Management Console snap-in). These management features include scheduling of updates, scheduling of scans and remote installation of Norton Antivirus client.

Any additional feature or helper program is another potential hole or point of failure and entails more information gathering and more testing. There is often a trade-off between functionality and security. A sensible approach is to narrowly define the functionality requirement from the server and to only install the minimal set features required to provide these requirements. Testing should feed back into this process, allowing the functionality requirements to be reassessed if they are impossible, difficult or expensive to secure.

In our case, we limited the server's functionality to the LiveUpdate Administrative Utility and installed none of the other features. Testing reinforced the decision by showing System Center's reliance on NetBios, a protocol that I prefer not to run on Internet-facing servers.

Note that the remainder of this paper will only discuss issue relevant to the features installed.

3.1.2 Patching the application

All applications have bugs, and most vendors provide fixes for these bugs. The first step to take when hardening an application is to check the vendor's website for patches. If a patch exists for your application, it can affect your testing & observations. The dilemma is whether to proceed with the un-patched application and retest, or to start with the patched application.

My recommendation is to read the documentation and decide if the patch is relevant to your environment. If the patch is relevant, patch the application and proceed with the testing; it is a luxury to test the un-patched application when you know that you will apply the patch.

N.B. The recommendation above only applies when installing new applications. Once the application is deployed, each individual patch has to be tested.

In our case, LiveUpdate is by it nature "self-patching" and the best way to insure that we are dealing with the latest release is to run LiveUpdate before testing. Note that there is a page on the Symantec web site that list the updates available for Norton Antivirus Corporate Edition: http://www.symantec.com/techsupp/enterprise/products/nav/nav-75-ce/files.html, but the only update listed is a link to the latest virus definitions.

3.2 Gathering information

To prepare for the hardening of a LiveUpdate server, we will gather information from the vendor, from various mailing lists and web sites as well as from direct observation. This information gathering exercise will allow us to insure that we can reasonably protect the application without breaking it.

3.2.1 The vendor documentation

The best place to start with any third party application is the documentation provided by the vendor. This obvious statement, often referred to by the RTFM acronym, is often overlooked.

In our case, Norton Antivirus Corporate Edition is sold with two administrative manuals: the System Center implementation guide and the Norton Antivirus Corporate Edition implementation guide. These manuals cover the configuration of a LiveUpdate Server in 12 pages [Symantec 2000a] with an emphasis on GUI-based management through Symantec System Center. All of the manuals are available in PDF format on the vendor's web site: http://www.symantec.com/techsupp/enterprise/products/nav/nav-75-ce/manuals.html. A short configuration document is also available on the site [Symantec 2002b].

For the sort of information required by inquisitive security people, there is a readme.txt file that covers the installation in much more details. The readme.txt file gives details on a variety of topics including the LIVEUPDT.HST file, silent retrieval mode, use of UNC path, logging and known issues. There is a LiveUpdate Administration README.TXT file available on the Symantec site [Symantec 1999] but the one distributed with the software is more up to date [Symantec 2001a].

Some good, but terse, information can also be found online in the Knowledge Base. The article below is a full quote from the Knowledge Base which illustrates the style and content:

```
Situation:
You want to configure a firewall for LiveUpdate to access
the Internet.

Solution:
LiveUpdate requires unblocked Internet access. If you are
using a firewall, then it must be setup to allow this, or
LiveUpdate will fail.

The following is the information that is needed to
configure your firewall to allow this access. For
instructions on how to enter these settings, read your
firewall's documentation or contact the manufacturer.

Ports, protocol, and the file information

- LiveUpdate requires access to ports 80 (HTTP), 21 (FTP)
and 443 (HTTPS).
- The protocol used is TCP.
- The file that accesses the Internet is Lucomserver.exe.
By default that file is located in C:\Program
Files\Symantec\LiveUpdate. [Symantec 2002a]
```

For our purpose, the important points to retain for the vendor documentation are:

- On the client side, the whole setup relies on a custom host file: liveupdt.hst
- The clients can use UNC paths, FTP or HTTP to retrieve updates
- The host file contains a username and password to log on to the server.
- The application consists of two main executables: luadmin.exe and luall.exe
- There is an option for silent package retrieval w/ luadmin.exe -silent.
- The administrative utility uses HTTP or FTP to retrieve updates from Symantec
- The executable that connects to the outside world is lucomserver.exe
- The LiveUpdate administrative utility can log event to the WIN2K event log

3.2.2 Observation part 1: server resources

There are a number of tools available to analyse executables and libraries, and these tools are extremely useful when hardening applications.

The first tool to turn to is Process Explorer from www.sysinternals.com. This tool allows you to view all the dlls and handles (files, registry, ports, threads) used by a process. The depends.exe utility from the WIN2K support tools can be called from Process Explorer and it provides additional information about an executable or dll. These two tools should provide you with a list of resources used by the application, this list will help you insure that the hardening, in particular, aggressive access control lists (acl) will not prevent access to a required resource.

From depends.exe and Process Explorer we can see which dlls are required by the executables, the example below shows the output obtained with "Save As" in Process Explorer (note that some columns have been omitted for clarity):

luadmin.exe:

Advanced Windows 32 Base API	C:\WINNT\system32\ADVAPI32.DLL
Common Controls Library	C:\WINNT\system32\comctl32.dll
Common Dialogs DLL	C:\WINNT\system32\COMDLG32.DLL
GDI Client DLL	C:\WINNT\system32\GDI32.DLL
Keyboard Language Indicator Shell Hook Extension	C:\WINNT\system32\indicdll.dll
Keyboard Support Functions	C:\Program Files\Symantec\pcAnywhere\awhk32.dll
LiveUpdate Admin Utility	C:\Program Files\LiveUpdate Administration\LuAdmin.exe
LiveUpdate Compatibility Module	C:\Program Files\Symantec\LiveUpdate\S32LIVE1.DLL
	C:\program files\liveupdate administration\SYMZIP.DLL
LZ Expand/Compress API DLL	C:\WINNT\system32\lz32.dll
Microsoft (R) C Runtime Library	C:\WINNT\system32\msvcrt.dll
Microsoft OLE for Windows	C:\WINNT\system32\OLE32.DLL
NT Layer DLL	C:\WINNT\system32\NTDLL.DLL
Remote Procedure Call Runtime	C:\WINNT\system32\rpcrt4.dll
Rich Text Edit Control, v3.0	C:\WINNT\system32\riched20.dll
Shell Light-weight Utility Library	C:\WINNT\system32\shlwapi.dll
Version Checking and File Installation Libraries	C:\WINNT\system32\version.dll
Windows 2000 IMM32 API Client DLL	C:\WINNT\system32\imm32.dll
Windows 2000 USER API Client DLL	C:\WINNT\system32\USER32.DLL
Windows NT BASE API Client DLL	C:\WINNT\system32\KERNEL32.DLL
Windows Shell Common Dll	C:\WINNT\system32\SHELL32.DLL
Windows Spooler Driver	C:\WINNT\system32\WINSPOOL.DRV
Wrapper Dll for Richedit 1.0	C:\WINNT\system32\riched32.dll

Process Explorer will also give you information about the security context under which the application runs. In particular, it will give you the user rights (privileges) under which the process is currently running.

Note that these tools do not always give you an exhaustive list because of the different ways programs can call each other under Windows. For best results you should have Process Explorer running while actively using the application. In our example we can see that when retrieving updates from the admin utility, the luadmin.exe process starts the luall.exe process but that the lucomserver.exe process appears under:
`System\SMSS.EXE\CSRSS.EXE\WINLONGON.EXE\SERVICES.EXE\svchost.exe`

Three other tools from www.sysinternals.com can help you further understand the resources used by an application, these tools are filemon.exe, regmon.exe and tokenmon.exe. These are respectively a file monitor, registry monitor and security token monitor. While Process Explorer gives you a snapshot of the resources used by a process, the three monitor tools can give you a "film" of the resources used by the application over a period of time.

In our example, we can see that lucomserver.exe saves files to a download directory before moving them to a temp directory. Then luadmin.exe moves them from the temp directory to the download directory specified in the admin utility GUI (D:\LiveUpdate). The output below from filemon.exe illustrates this (output edited for clarity):

```
LUCOMS~1.EXE:2784    IRP_MJ_WRITE    C:\Documents and Settings\All
Users\Application Data\Symantec\LiveUpdate\Downloads\enlu170a.x86
SUCCESS      Offset: 647168 Length: 4096

LUCOMS~1.EXE:2784    IRP_MJ_CREATE    C:\program files\liveupdate
administration\temp\enlu170a.x86
SUCCESS      Attributes: Any Options: Open

LuAdmin.exe:548      FASTIO_READ      C:\program files\liveupdate
administration\temp\enlu170a.x86
SUCCESS      Offset: 1835008 Length: 65536

LuAdmin.exe:548      FASTIO_WRITE     D:\LiveUpdate\enlu170a.x86
SUCCESS      Offset: 196608 Length: 65536
```

For in-depth explanations of the sysinternals tools and the inner workings of Windows 2000 see [Solomon and Rusinovich 2000].

3.2.3 Observation part 2: network resources

To analyse network utilization, the best tools are fport from www.foundstone.com and a packet capture tool such as tcpdump from www.tcpdump.org, windump windump.polito.it or Ethereal from www.ethereal.com.

With fport we can determine on which port(s) applications are communicating. In our case the LiveUpdate Administration does not listen on any port, but we know that we will have to set up an FTP server later to service the clients. When lucomserver.exe connects to the Symantec servers it uses the http protocol or, if http is unavailable (e.g. blocked at the firewall), the ftp protocol. The fport output below illustrates both cases (multiple outputs truncated for readability):

```
3272  LUCOMS~1        ->  2559  TCP
C:\PROGRA~1\Symantec\LIVEUP~1\LUCOMS~1.EXE

3272  LUCOMS~1        ->  2560  TCP
C:\PROGRA~1\Symantec\LIVEUP~1\LUCOMS~1.EXE

3368  LUCOMS~1        ->  2766  TCP
C:\PROGRA~1\Symantec\LIVEUP~1\LUCOMS~1.EXE
```

The fport utility does not fully show network connections; the remote ports are absent from the output above and it is impossible to distinguish the http traffic from the ftp traffic. To obtain the full picture, protocol analysers are needed. With tcpdump, we can see that port 2559 was used for the ftp control connection, port 2560 for the ftp data connection, and that port 2766 was used in the http transfer (the output below has been filtered for readability):

```
20:39:08.995089 192.168.0.100.2559 > 206.204.212.72.ftp: S
824620526:824620526(0) win 16384 <mss 1460,nop,nop,nop,nop> (DF)

20:39:09.106642 206.204.212.72.ftp > 192.168.0.100.2559: S
3865551757:3865551757(0) ack 824620527 win 64240 <mss 1460> (DF)

20:39:10.032193 192.168.0.100.2560 > 206.204.212.72.50613: S
824884649:824884649(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)

20:39:10.144105 206.204.212.72.50613 > 192.168.0.100.2560: S
1571878128:1571878128(0) ack 824884650 win 64240 <mss 1460> (DF)

20:59:27.953497 192.168.0.100.2766 > 193.38.108.221.http: S
1107586574:1107586574(0) win 16384 <mss 1460,nop,nop,sackOK> (DF)

20:59:27.996735 193.38.108.221.http > 192.168.0.100.2766: S
2206125024:2206125024(0) ack 1107586575 win 8760 <mss 1460> (DF)
```

With ftp it is always good to check if the application supports passive ftp. In normal ftp, the data connection is initiated by the server and always uses port 20 on the server and a dynamically assigned port on the client. In passive ftp, the client initiates the data connection and uses dynamically assigned ports on both ends of the connection. Passive FTP is more secure, principally because it does not leave the client open to connection from a known port [Zwicky, Cooper and Chapman 2000 pp.455-470]. In the present case Live Update used passive ftp by default.

3.2.4 Vulnerabilities

After studying the vendor's documentation and observing the recourse usage of the application the last step is to search for previously released vulnerabilities. Vulnerabilities are constantly being discovered and to get up to date information, we have to turn to the Internet. Good sources for published vulnerabilities are: www.cert.org and the BUGTRAQ mailing list archived at www.securityfocus.com. The Common Vulnerabilities and Exposures (CVE) project www.cve.mitre.org is also useful as it gives a single name to vulnerabilities and avoids confusion if a vulnerability has been published in multiple places or under different titles. Typing in the name of your application and the name of the various executables in your favourite search engine can also dig up some valuable information.

In the current case, we found the main security issue with the LiveUpdate mechanism to be that clients are relying on DNS to find the Symantec server. You could theoretically get a LiveUpdate client to download and install files from an arbitrary server by poisoning DNS entries or spoofing (DNS or Symantec). Symantec has addressed this issue in LiveUpdate version 1.6 and above. In these versions, the LiveUpdate utility verifies that the update have been digitally signed by Symantec before installing them. Note that there is no publicly available description of the method used to sign the files, nor is there a published cryptanalysis of the method used.

Although the digital signatures prevent the installation of "fake" updates, there is a possible Denial of Service (DoS) attack. Phenoelit www.phenoelit.de described such a scenario in their Advisory #0815. The attack starts with the spoofing of the Symantec site. By spoofing the site, the attacker can force the application to download an arbitrary file. The digital signature insures that the file will not be installed, but the check occurs only after the file has been downloaded. Hence by using a very large file, the attacker can prevent the application from downloading the real updates, and overload the victim's network and server leading to some denial of service.

> […] by specifying a large file location on the Internet, a scheduled LiveUpdate session in a medium sized company will lead to network degradation and outages due to the large amount of traffic generated […] [Phenoelit 2001]

Symantec has a response to this advisory on their site which addresses the potential DoS specifically:

> Symantec's LiveUpdate 1.6 could potentially be temporarily affected by the DoS scenario depicted by the Phenoelit group however, only a small percentage of a very large user base could potentially be impacted to any degree as the spoofing or redirection would, by it's very nature, be limited to a local Internet area/region. [Symantec 2002c]

DNS abuse is a broad issue that goes beyond LiveUpdate and most organizations have to evaluate and manage the risks associated with their DNS infrastructure; if the current provisions are satisfactory, then they should be satisfactory for the LiveUpdate server as well.

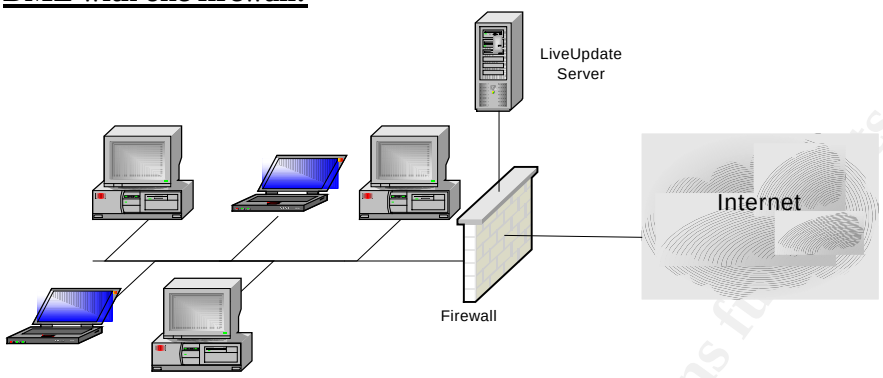
This advisory also makes a good case for implementing the LiveUpdate Server; if the attack occurs the server will be affected, but at the same time it will protect the clients machines from the DOS.

4 Network Topology

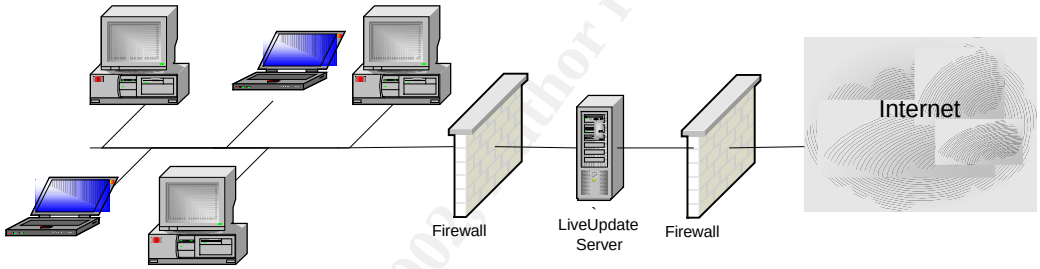
Hardening servers and applications is not done in a vacuum. The network architecture influences the way in which a bastion host is built. Ideally, the network architecture and host configuration will complement each other to provide defence in depth.

The LiveUpdate Server needs to have access to the Internet and be accessible from the internal network. The server needs to be firewalled from the Internet and firewalled from the internal network. There are two basic designs to accomplish this and each design can incorporate NAT.

DMZ with one firewall:



DMZ with tiered firewalls:



The single firewall approach is cheaper because it requires less hardware, but that is its only advantage. The tiered approach is more secure and easier to manage:

- With a tiered approach the inside network is still protected if the first firewall is compromised.
- The tiered approach allows us to keep inside-facing and outside-facing traffic completely segregated.
- The tiered approach leads to simpler rules on the firewalls, rules that are less likely to be mis-configured and require less processing power.

In the present case we leveraged an existing tiered infrastructure.

5 Hardening the server & application

5.1 Operating System

5.1.1 OS selection

The first step in building the LiveUpdate server is to select the OS. This choice is limited as Symantec NAV Corporate Edition only supports Windows NT 4.0 and Windows 2000.

A good practice when selecting an OS for a bastion/hardened host is to use a "tried and tested" operating system with all of the available patches. The better tested of the two available OS is NT 4.0, but in the present case, we selected Win2K for its greater stability and enhanced features (in particular the IPSec functionality).

5.1.2 Patching

Microsoft has made patching a lot more manageable with the introduction of the Hotfix Checker (hfnetchk.exe). This command line tool checks the following for patches:

- Window NT 4.0
- Windows 2000
- Windows XP
- IIS 4.0
- IIS 5.0
- SQL Server 7.0
- SQL Server 2000
- Internet Explorer 5.01 or later

The tool relies on an xml file (mssecure.xml) that is constantly updated. By default, the tool will try to download a compressed version of this file from the Microsoft site every time it runs. The tool then compares files, file versions and registry settings between the xml file and the target machine to determine if a patch has been properly applied [Microsoft 2001a].

With the LiveUpdate server, we ran hfnetchk.exe after installing the OS, after patching the OS and after installing the LiveUpdate Administrative utility. This insured that installing the application did not affect the patches.

5.2 The FTP server

5.2.1 Choosing the FTP server

The only service that is needed on the server is FTP. The most important features needed for the FTP service are: the ability to limit the interface on which the service is provided (our sever is dual-homed), the ability to run a low privileged account, and the ability to limit the directories to which the clients has access. The Windows 2000 ftp service available as part of IIS 5.0 has these features, and it relatively well tested and documented.

IIS has had more than its fair share of vulnerabilities, but the ftp service has had relatively few. In choosing IIS, we also considered the existing skill set of the system administrators who were all very familiar with IIS.

5.2.2 Configuring the FTP service

There are four main parts to configuring the ftp service: ip addresse(s), home directory, authentication and logging. Most of the configurations can be done through the properties of each FTP site (virtual server).

We chose to create 2 virtual ftp servers on 2 ip addresses on the inside network of our tiered DMZ. The two ip addresses are crucial to the change control procedure for updates (see section 6.3.1).

As a protection against directory traversals attacks, the ftp home directories for each virtual server were set up on a partition separate from the system partition. This is the partition that is available (read-only) through the ftp service.

The ftp service can let users log in anonymously or log in as a "real" user. The ftp protocol sends usernames and passwords in clear so using real accounts should never be allowed. The anonymous logging is not truly anonymous; it uses a local user account. By default this account is IUSR_[computername] and it is shared with the World Wide Web Service. The best approach is to configure a new ftp account for anonymous logins.

Logging for FTP, like logging for the web service is done in W3C Extended Web Format which is a standard plain text format. The fields to log, the log file location, and the log file rotation can all be configured through the properties window.

In addition, there are two security related registry keys to check under HKLM\System\CurrentControlSet\Services\MSFTPSVC\Parameters:

- EnablePortAttack=0 (prevents ftp data connection from ports below 1024)
- AllowGuestAccess=0 (prevents guest login)

Note that these keys are set to 0 by default under Windows 2000 but that guest access was allowed by default in Windows NT [Microsoft 1996].

5.2.3 FTP user account & rights

A specific user was set up for the ftp anonymous account. The login and password were hardcoded respectively to anonymous and anonymous@LiveUpdate in the configuration file that resides on the client on the LiveUpdate clients. The ftp user was only given read permissions on the ftp partition (section 5.2.2) and was explicitly denied access to the other (system) partition.

5.2.4 Configuring the FTP service account

One of the most important precautions to take when protecting server applications on any platform is to run the service/daemon under a restricted account. This limits the ability to use exploits (buffer overflows in particular) to take full control of a machine. By default the ftp service, like all windows services, runs as LocalSytem, an account that has full access to machine. The only restriction placed on LocalSystem is that the account cannot be used to log on over the network. We need to create a restricted account to run the ftp service.

Creating services account under Windows 2000 is not a trivial exercise:

> It would take significant amount of work and documentation to figure out the account configuration that could be used to run each of the different services, which is probably the reason that Microsoft has not done it and just uses LocalSystem. [Cox and Sheldon 2001 p. 664]

Although the ftp services appears separately in the list of services, it is a monolithic executable (inetinfo.exe) that also provides four other services. When configuring our ftp service account, we need to configure it for the other inetinfo.exe services: the World Wide Web Publishing service, the Network News Transport Protocol (NNTP) service, the Simple Mail Transport Protocol (SMTP) service and the IIS Admin Service. This is not required if the other services are disabled.

Our ftpdaemon account is a slight improvement over LocalSystem, it is a member of the Administrator group, but has the following user rights assignments:

- Deny access to this computer from the network
- Deny logon as a batch job
- Deny logon locally

Furthermore, all users and groups were removed from the following user right assignments:

- Bypass Traverse Checking
- Create a pagefile (once the pagefile have been configured)
- Create a token object
- Create a permanent shared object
- Debug programs (This is required for a number of sysinternals tools but should not be enabled on production machines)
- Force shutdown from a remote system
- Increase quotas
- Increase scheduling priority
- Load and unload device drivers

Finally, the ftpdaemon account was denied access to the following executables (adapted from [Norberg 2001]):

arp.exe
at.exe
cacls.exe
cscript.exe
cmd.exe
ftp.exe
ipconfig.exe
net.exe
netstat.exe
nslookup.exe
ntbackup.exe
ping.exe
rdisk.exe
regedit.exe
rededt32.exe
route.exe
runonce.exe
syskey.exe
tftp.exe
tracert.exe
winmsd.exe
wscript.exe
xcopy.exe

5.3 Networking

5.3.1 SYN flood protection

The firewalls protect the server against SYN floods, but as an additional precaution, SYN flood protection was enabled on the server by configuring the following key [Microsoft 2001b]:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters
With the value:
SynAttackProtect=2

5.3.2 ICMP Redirect

To protect the server's routing table, the ICMP redirect was disbled by setting the following key to 0 [Norberg 2001]:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\EnableICMPRedirects

5.3.3 IpSec

Following the defence in depth principle, we decided to back up the IP filtering done by the firewall with Windows 2000 IPSec policies. IPSec policies are preferable to NT 4.0 TCP/IP filters, especially on multi-home machines, because they are not applied in a blanket fashion to all adapters.

In our example, we used IPSec to drop all traffic coming in on tcp 21 on the outside-facing ip.

5.4 LiveUpdate configuration

Compared to the configuration of the OS and FTP service, the configuration of the LiveUpdate administrative utility is straightforward.

5.4.1 Configuring the retrieve options

The Management Utility needs to be configured to retrieve the relevant products and the relevant languages. It should also be configured to retrieve new updates only (Tools|Option . . .). The download directory is the staging/testing directory on the ftp partition created in section 5.2.2.

5.4.2 Creating the host file

Creating the LIVEUPDT.HST file is easily done through the Host File Editor GUI. The fields to specify are Name, IP Address, Type (FTP), Subnet and Subnet Mask. Our configuration requires two host files: one pointing to the test ip address and one pointing to the normal ip address see section 6.3.1 for details.

5.4.3 Scheduling the updates

The updates were scheduled through the Schedule Tasks applet in the control panel. The command was: luadmin.exe –silent.

6 Operational Security

Bad operational security can ruin all of the hard work put into hardening an application. Day-to-day operations can introduce weaknesses in your security infrastructure, but good procedures can counter this eroding effect.

6.1 Physical security

Physical security is the fist step towards operational security. No change control procedures or hardened configuration can save you if your server is carted away by a thief or melted in a fire.

6.2 Remote Server Management

Since the environment in which we installed our server is manned 24x7x365, the server is connected to a KVM switch, and the anticipated maintenance was low, we did not implement any remote management solution.

Note that remote management software should be analysed in the same fashion as any other application and hardened appropriately. The same logic applies to KVM switches and terminal servers/concentrator available over the network. If the remote management solution includes a modem, either as a primary tools or an out of band backup, the modem & modem software should be audited.

6.3 Change Control

Security is a process and change control is a crucial part of it. Over time, changes can intentionally or unintentionally weaken the security posture of servers.

6.3.1 Signature and executable update procedure

There is a significant risk associated with a fully automated LiveUpdate setup. A buggy update to one of the Symantec executable or virus signatures could affect client machines.

To test updates, we use 2 virtual ftp servers on the LiveUpdate server. One ftp server provides access to the updates that are automatically downloaded from the Symantec website to a restricted pool of representative staging machines. If no issues are detected on the staging machines, the updates are copied to the home directory of the second virtual server. The second virtual directory is the source of update for all the other client machines.

6.3.2 OS Patch management

Once the server has been deployed, OS patches need to be carefully selected and tested. To help in testing without maintaining a large number of servers we use vmware (www.vmware.com) unless the server runs exotic hardware.

6.4 Backups & "Golden Image"

All rules have exceptions and our LiveUpdate server is the exception to "Thou shall make regular backups". The server stores no crucial data, and all that is required to get back to an operational state is a "golden image". To speed up recovery, a new image should be made after each OS patch.

7 Avenues for further development

The signature/encryption mechanism that Symantec utilizes to check the validity of the downloads is not documented. There are therefore no independent guarantees that validate the cryptography and implementation of this feature.

The ftp service account (ftpdaemon) should be locked down further by removing it from the Administrator group.

8 References

Cox P. and Sheldon T. 2001. Windows 2000 Security Handbook. Osborne/McGraw-Hill, Berkeley, CA.

Microsoft 1996. IIS FTP Service Registry Parameters. http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q147621& Last accessed 19th August 2002

Microsoft 2001a. Microsoft Network Security Hotfix Checker (Hfnetchk.exe) Tool Is Available. http://support.microsoft.com/default.aspx?scid=kb;en-us;Q303215&sd=tech Last accessed 16th August 2002.

Microsoft 2001b. HOW TO: Harden the TCP/IP Stack Against Denial of Service Attacks in Windows 2000. http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q315669& Last accessed 20th August 2002

Norberg S. 2001. Securing Windows NT/2000 Servers for the Internet. O'Reilly & Associates, Sebastopol, CA.

Phenoelit 2001. Phenoelit Advisory #0815 http://www.phenoelit.de/stuff/LiveUpdate.txt Last accessed 13th August 2002.

Solomon D. A. and Rusinovich M. E. 2000. Inside Microsoft Windows 2000 Third Edition. Microsoft Press, Redmond, WA.

Symantec 1999. README.TXT File for LiveUpdate Administration Program LUADMIN.TXT. http://service4.symantec.com/SUPPORT/sharedtech.nsf/d3c44a1678bd8f45852566aa005902cb/17d1cb88c94baee2882565da00810666?OpenDocument Last accessed 13th August 2002.

Symantec 2000a. Norton Antivirus Corporate Edition 7.5 Implementation Guide. Symantec Corporation, Cupertino, CA.

Symantec 2000b. Symantec System Center 4.5 Implementation Guide. Symantec Corporation, Cupertino, CA.

Symantec 2001a. LiveUpdate Administration Utility (Version 1.5.3) Readme.txt, Symantec Corporation, Cupertino, CA.

Symantec 2002a. Settings needed to configure your firewall for LiveUpdate. http://service2.symantec.com/SUPPORT/sharedtech.nsf/d3c44a1678bd8f45852566aa005902cb/5b395e9b7aeb5b4388256b9800798630?OpenDocument Last accessed 13th August 2002.

Symantec 2002b. How to configure the LiveUpdate Administration Utility. http://service2.symantec.com/SUPPORT/sharedtech.nsf/d3c44a1678bd8f45852566aa005902cb/f81b8983d1b505c7882569700061f53d?OpenDocument Last accessed 13th August 2002.

Symantec 2002c. Security Response Symantec LiveUpdate 1.4 through 1.6 vulnerability. http://securityresponse.symantec.com/avcenter/security/Content/2001.10.05.html Last accessed 13th August 2002.

Zwicky E. D., Cooper S. and Chapman D. B. 2000. Building Internet Firewalls, 2nd Edition. O'Reilly & Associates, Sebastopol, CA.