Global Information Assurance Certification Paper

HOWTO: Logon Scripts as Anti-Virus Tools

Brad Blauvelt

GSEC Version 1.4 Option 1

A father was attempting to explain viruses, trojans and worms to his child. "It's like this, honey," he said. "Computer viruses sneak in when nobody's watching, mess up a lot of stuff they shouldn't, and reproduce as much as they can." After a moment's thought, the child replied. "But daddy, you do all those things too. Are you a computer virus?"

Abstract

Any personal resemblance to viruses notwithstanding, Network Administrators (NetAdmins) often have their hands full dealing with these insidious threats. From oldies but baddies like Michelangelo or Friday the 13th to such newer threats as Nimda and Klez, NetAdmins need complete Anti-Virus (AV) toolkits.

This paper shares how-to info about an easy-to-use, but often overlooked, tool. Through the use of Network Logon Scripts, NetAdmins can effectively manage their users' AV installations. Sample "batch file" scripts are included. They can check for the existence of AV software on client computers, install AV software if needed and automatically push AV data file (DAT) updates to client. While doing so they can log the results to a central server for a NetAdmin to peruse. If virus infections or hoaxes should occur, Network Logon Scripts can locate the infected computers, notifying user and/or NetAdmin as needed.

The NetAdmin's Anti-Virus (AV) Toolkit

The threat vectors for Viruses, Trojans and Worms aren't discussed here. Thinkquest.org's "Is it A Virus or What" shares a brief explanation.[1] For the sake of brevity, all such threats in this paper are referred to as "viruses."

As viruses work in multivariate ways, a corporate AV toolkit must also be multifaceted. Many of the same security tools and procedures that protect the network from hacking are also helpful in preventing virus infections.

- A firewall can stop many viruses when properly configured.[2] Some firewall solutions also include integrated virus scanning.
- Keeping server and client software patches current reduces vulnerability.
- Intrusion Detection Systems can detect virus activities. Snort's Signature File #795, for example, can detect *.txt.vbs virus types.[3]
- Blocking or disallowing peer-to-peer (P2P) file sharing programs should be considered. See Billy Evans excellent SANS paper outlining the risks.[4] Oofle.com lists specific P2P sites and how to block them at the firewall.[5]

- Security Policies covering virus prevention, infection response and virus hoaxes are vital.[6]
- Server-based AV solutions, such as a Sendmail milter[7] or Microsoft Exchange AV software,[8] can stop infections at the Message Transfer and/or Mail Delivery Agent. Proxy servers can also be configured to scan content for viruses.
- Network Gateway AV scanning is useful, which can be software-based or use dedicated AV appliances.[9]
- A properly configured and current "Real-Time" (Memory-Resident) AV solution on every system is essential.[10]
- Automated management of client AV software will keep clients current.

Real-Time AV is unpopular with some NetAdmins and users, as it could be likened to a 350-pound bodyguard, who inhabits your living room all day, eating up your food. Indeed, AV software can gobble system resources and negatively impact performance. But, in the same way that a home's locks, barred windows and alarms won't stop a determined intruder, no network-based AV will fully protect client computers. One floppy disk, CD, PDA or modem is all it takes to circumvent every non-client-based AV. Every client computer must have its own bodyguard.

Managing the AV on all of the clients can be extremely time-intensive for a NetAdmin. Oftentimes the clients often aren't sufficiently protected. The Klez infection was a good case in point. If the client's AV DAT files were even moderately current, they would have been safe. As Klez infected over seven percent of all computers,[11] many clients were poorly protected. NetAdmins spent countless hours on the sneaker-net (physically going from computer to computer), to manage the outbreak.

Some AV manufacturers provide tools to manage client installations, such as McAfee e-Policy Orchestrator[12] and Norton Anti-Virus CE Server.[13] They, as well as similar products, are extremely useful. But for NetAdmins who can't (or won't) purchase such products; Network Logon Scripts can serve as a viable alternate. Scripts are capable for a substantial amount of AV management.

Network Logon Scripts as AV Managers

Network Logon Scripts commonly reside in the NETLOGON share on Windows Domain Controllers, residing by default in the %systemroot%\System32\Repl\Import\Scripts directory on Windows NT Servers, and in the %systemroot%\SYSVOL\sysvol*domain name*\Scripts directory on Windows 2000 Active-Directory Domain Controllers. Similar scripts can also be run from Novell Netware and UNIX/Linux Samba shares.

Logon Scripts can be a batch file (*.bat or *.cmd), executable, or a procedure created in Visual Basic, Windows Script Host or Java Script. For easiest readability, the samples in this paper are all batch files.

This paper is not a primer on batch file scripting. Information on batch file programming is widely available on the Internet, in the Open Directory Project's list, for example.[14] The specific syntax used in the following samples is explained for novice batch file creators. Experienced batch file programmers will find the comments within the samples sufficient.

These sample batch files are written in "pure" batch language, without using third-party tools or Windows NT/2000/XP Command Extensions. Freeware tools such as ECHON, GAWK or SED, as suggested by Professor Timo Salmi,[15] can make batch files even more powerful. Batch enhancements such as freeware software KiXtart[16] or commercially available Script Logic[17] can simplify logon script creation.

These batch file samples assume the client computers are using Symantec's Norton Anti-Virus and/or Network Associates' McAfee VirusScan. This paper doesn't presume that they are in any way superior to other products. Numerous providers have solutions that may work better for many users. Jacqueline Castelli's paper "Choosing Your Anti-virus Software" shares criterion for finding the appropriate package.[18] So now that we're (hopefully) AV politically correct, we'll "goto" the first sample.

Sample script #1 – Check for AV Software & Generate Report

This is a very simple, yet useful set of lines that could be added to a logon script.

```
@echo off
:: AVfiles.bat
::
:: Check for McAfee or Norton Anti-Virus Software.
:: If they don't exist, write a log entry on a server.
::
:: Check for McAfee executable. If found, jump to AVokay label.
if exist "c:\program files\network associates\virusscan\avsynmgr.exe" goto avokay
::
:: Check for Norton executable. If found, jump to AVokay label.
if exist "c:\program files\norton antivirus\navapw32.exe" goto avokay
::
:: Create log file entry if no AV software exists.
@echo %computername% does not have McAfee or Norton installed >>
\\myserver\logdir\noav.log
::
:: AVokay Label - If AV software exists, above lines will jump to this label
:avokay
```

In this first sample, the script checks at logon for McAfee and Norton AV executables. If they don't exist, it writes a file to a central location. A NetAdmin can later review it there.

The first @ECHO OFF command simply keeps itself and the following lines from appearing in the logon box window. The lines beginning with double colons (::) are used as remarks to make the script more readable. Some NetAdmins use the REM command for remarks, which slows batch processing slightly. The Command Processor sees REM as a command it must process. Double-colons are seen as invalid labels, so they are skipped. Double-colons are therefore the better way of entering comments.

The IF EXIST lines check for specific variables. The quotation marks (" ") preceding and following the paths are needed for non-8.3-type directory or file names (names containing spaces or long names) They must be simple quotes, not smart quotes.

The GOTO command tells the script to "go to" a label, which always begins with a single colon (:). When the first IF EXIST line finds McAfee VirusShield, GOTO AVOKAY will skip to the AVOKAY label, ignoring any lines in between. If not, the next lines will be processed in turn.

If the second IF EXIST line fails to find Norton AV, it won't go to the label either, and the "@ECHO %COMPUTERNAME%…" command will process.

The last @ECHO command line is used to write a report. It uses an environment variable named %COMPUTERNAME% to report which computer is involved. Those who aren't familiar with environment variables can type the word SET while in a Command-Prompt (DOS) window. The lines that appear there are all environment variables.

Following the %COMPUTERNAME% environment variable are the words "does not have McAfee or Norton installed." This is free-form text. When NetAdmins create their own scripts, it can contain any phrase. Though there are limits to how long a command line can get, it shouldn't be a problem as long as the text isn't overly verbose.

The greater-than symbols ">>" redirect the output text to somewhere else, rather than just echoing it to the console. In this case, it's redirecting it to a server named *myserver* into a Windows share named *logdir* to a file named *noav.log*. The double greater-than symbol (>>) is used instead of a single one (>) as it will not only create the output file, but will append to an existing one.

Here's what *noav.log* looks like on the destination server. In this example the computers named MOE-SHUDDUP, LARRY-NYUCK and CURLY-WISEGUY were lacking the looked-for AV software:

Another possible use for a similar script is one that checks for the existence of insecure/dangerous applications. To write a script to find them, the NetAdmin needs to know the specific directory paths and/or file names, and then set the script variables as needed. An example of a similar script is below - Sample Script #7, a method for finding virus-infected computers.

Sample Script #2 – Check Registry for Real-Time AV & Generate CSV

This is a more sophisticated test of the client AV. NetAdmins may need to know if the client computers are running “Real-Time” AV as well. Also known as “Memory-Resident” or “TSR,” Real-Time AV is always on the lookout for viruses. In this sample the Window Registry is parsed for the appropriate AV startup commands. It then sends the output to the server in a “Comma-Separated-Value” (CSV) format, importable into many applications.

```
@echo off
:: RealTime.bat
::
:: Check the Windows Registry for the existence of
:: Real-Time startups for Norton and McAfee AV programs.
:: If they don't exist, write to a .csv file on the server.
::
:: Check to see if Operating System is Windows NT or later.
:: If so, jump past the Windows 9x Section, directly to the Windows NT/2000/XP Section.
if %os%==Windows_NT goto winnt
::
:: Windows 9x Section.
::
:: Test for Norton Anti-Virus Startup.
::
:: Check for existence of executable. If it doesn’t exist, jump to Test2 label to scan for McAfee.
if not exist "c:\program files\norton antivirus\navpw32.exe" goto test2
::
:: Run Registry editor and export the Run key to a file named {1}.txt.
start /w regedit /e {1}.txt
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
::
:: Search the exported registry key for Norton startup executable.
type {1}.txt | find /i "NAVAPW32.exe" > nul
::
:: If NAVAPW32.exe is found, jump to Done label.
```

```
:: Otherwise, jump to Test2 label to scan for McAfee.
if errorlevel==1 goto test2
if not errorlevel==1 goto done
::
:: Test2 label – if Norton AV is not found, script will jump to this label.
:test2
::
:: Test for McAfee Anti-Virus Startup.
::
:: Check for existence of executable. If it doesn't exist, jump to Report label.
if not exist "c:\program files\network associates\virusscan\avsynmgr.exe" goto report
::
:: Run Registry editor and export the RunServices key to a file named {2}.txt.
start /w regedit /e {2}.txt
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices
::
:: Search the exported registry key for Norton startup executable.
type {2}.txt | find /i "AVSYNMGR.EXE" > nul
::
:: If AVSYNMGR.EXE is found, jump to Done label.
:: Otherwise, there's no Win9x startup, go to Report label.
if errorlevel==1 goto report
if not errorlevel==1 goto done
::
:: WinNT label – if Operating System is Windows NT or later, script will jump to here.
:winnt
::
:: Windows NT/2000/XP Section.
::
:: Test for McAfee VirusScan Service.
::
:: Check for existence of executable. If it doesn't exist, jump to "test4" label.
if not exist "C:\Program Files\Common Files\Network Associates\McShield\Mcshield.exe" goto
test4
::
:: Run Registry editor and export the McShield key to a file named {3}.txt.
start /w regedit.exe /e {3}.txt
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\McShield
::
:: Search the exported registry key for McAfee startup executable.
type {3}.txt | find /i "Mcshield.exe" > nul
::
:: If McShield.exe is found, jump to Done label, otherwise, go to Test4 label.
if errorlevel==1 goto test4
if not errorlevel==1 goto done
::
:: Test4 label – if McShield Service not found, script will jump to this label.
:test4
::
:: Test for Norton AntiVirus Application Service.
::
:: Check for existence of executable. If it doesn't exist, jump to Report label.
if not exist "C:\Program Files\NavNT\NAVAP.sys" goto report
::
:: Run Registry editor and export the McShield key to a file named {4}.txt.
```

```
start /w regedit.exe /e {4}.txt
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NAVAP
::
:: Search the exported registry key for Norton Anti-Virus Application sys file.
type {4}.txt | find /i "NAVAP.sys" > nul
::
:: If NAVAP.sys is found, jump to Done label, otherwise, jump to Report label.
if errorlevel==1 goto report
if not errorlevel==1 goto done
::
:: If no Real-Time AV executables and/or startup commands are found, generate report.
::
:: Report label. If search conditions above fail, script will jump to here.
:report
::
:: Create %date_% environment variable for use in report.
echo.exit | %comspec% /k prompt @set date_=$d$_rem > {4}.bat
call {4}.bat
::
:: Write the report to the server using the %date%_ environment variable.
@echo %date_%,%username%,%computername%,No AV Service >> \\myserver\logdir\noav.csv
::
:: Done label. If AV services exist, above lines will jump to this label.
:done
::
:: Cleanup - delete temporary files and environment variable created above.
del {?}.*
set date_=
```

Though the lengthy syntax in this batch file may appear intimidating, it's actually quite simple. Like Tinker Toys® or Lego Blocks®, all batch files are constructions of little commands, one building on the other. Taken step by step, each piece is easily understandable.

The script begins by using IF to check the version of the Operating System currently running. %OS% is an environment variable, which on NT and later systems will equal "Windows_NT" (case sensitive). This command line saves time in the batch file. When the IF condition is met, the script will GOTO the WINNT label.

The double-equals (==) in the command line are required. If the same command line were manually entered in a Command-Prompt (DOS) window, it would only need to be a single equals (=), but in batch files, it's necessarily doubled.

The script then uses IF NOT EXIST to check if the executable is actually present. If not, it won't bother checking the Windows Registry for the startup command, and will instead GOTO a label to perform the next test or to write the report.

START /W "starts" a program as a separate process and waits for it to run. In this case, it's starting REGEDIT.EXE, the Windows Registry Editor.

REGEDIT is the real magic in this batch file.[19] In this example it's exporting (/E) its results to temporary text files named "{*number*}.txt." The Norton Anti-Virus 9x test, for example, uses a temporary file named {1}.TXT.

The Windows Registry keys of "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run" and "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices are common startup locations for memory resident programs. In this sample, the startups for Norton and McAfee are found here on Windows 9x systems. Windows NT and later systems normally load Real-Time AV as Services, located under the "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services*servicename*" key.

After exporting the appropriate Registry key, the batch file will TYPE (output) the {*number*}.txt file. While doing so, it "pipes" (|) the output through the command FIND /I. Piping is a method of making one program act upon the results of another. In this case it's set to find a specific string of text in the output. FIND /I "ignores" the case, so capital letters and lower case appear to be the same. For Norton AV on 9x systems, FIND is looking for the text string of *NavW32.exe*. The results are redirected to nowhere (> nul).

When programs finish running, they automatically generate "errorlevels." Errorlevels are normally ignored, but in this script, they're used. If FIND succeeds in finding the text string, it will return an errorlevel of 0 (zero).

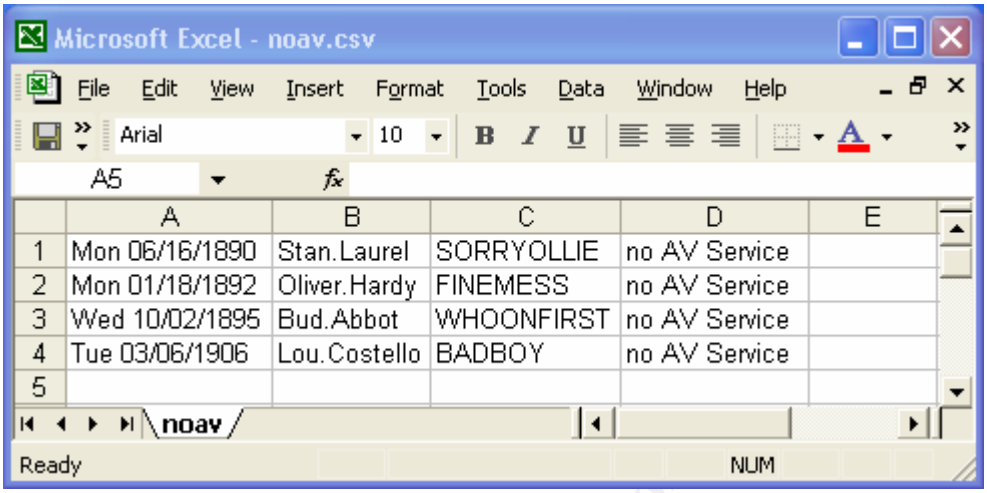
The script will then use the errorlevel – when the IF ERRORLEVEL is 1 (or higher), the script will skip to the next test's LABEL. It then uses IF NOT ERRORLEVEL to check if the errorlevel is not 1 (is 0). If so, it will skip all other tests and go directly to the DONE label.

If all of the tests fail and no Real-Time AV is found, the script will generate a report. It begins by creating a temporary environment variable named DATE_. The usage of the PROMPT command to set the date is a neat trick discovered by Tom Ladevas, explained on his web page.[20]

The report "echoes" (@ECHO) several environment variables, separated by commas, redirected (>>) to a server file. In this case, the file is named *noav.csv*.

Here's how *myserver**logdir**noav.csv* appears when opened in Excel. The first column is from the %DATE_% environment variable and is the date of the user logon. The second column is the %USERNAME% logon name and the third is the %COMPUTERNAME% variable. The fourth column contains the free-form

words set in the batch file. The dates in this sample spreadsheet are obviously bogus, as they're actually these "users" birthdays.



This sample batch file checks for Norton and McAfee Real-Time startups in Win 9x and NT or later desktop systems. For real-world usage, adaptations are needed to include all client AV permutations. Other uses for a similar batch file would be to search for undesirable startup procedures. An example is below in Sample Script #8, which searches for a virus startup in the Registry.

Sample script #3 – Check for AV Software & Install McAfee

It's also possible for the logon script to install AV software if none exists. In this sample it checks to see if Norton or McAfee are installed. If they aren't, it will install McAfee, and write to a log file on the server indicating that it did.

```
@echo off
::SetupAV.bat
::
:: Check for McAfee or Norton Anti-Virus Software.
:: If they don't exist, install McAfee from server share to client.
::
:: Look for McAfee executable. If found, jump to avokay label.
if exist "c:\program files\network associates\virusscan\avsynmgr.exe" goto avokay
::
:: Look for Norton executable. If found, jump to avokay label.
if exist "c:\program files\norton antivirus\navapw32.exe" goto avokay
::
:: Setup McAfee VirusScan. Send Setup log file to server.
\\myserver\install\mcafee\setup.exe ADDLOCAL=ALL /LSCRIPT /LE+
\\myserver\logdir\mcsetup.log /I
::
:: AVokay Label - If AV software exists, above lines will jump to this label.
:avokay
```

The syntax in this script is similar to Sample Script #1. The only difference is the "setup.exe" command line. SETUP.EXE is run from the server named *myserver*, from a Windows Share named *install* and a subdirectory named *mcafee*.

The sub-command of ADDLOCAL=ALL tells setup to install everything. /LSCRIPT tells setup that it is running from a logon script, /LE+ tells it to log everything, and append the results to the *\\myserver\logdir\mcsetup.log* file. The /I switch tells setup to begin the install. Other command-line setup options can be found in the McAfee VirusScan Administrator's Guide.[21]

Keeping the Client AV Data Files (DATS) Current

Most client AV software packages contain functionality to schedule their DAT updates. But in some circumstances, a NetAdmin may need to proactively push updated DATS. If there is a rampant new virus infection and the NetAdmin needs to get the DATS out quickly, they may need to become pushers.

The following two scripts show ways of using batch files to update client AV DATS.

Sample script #4 – a Poor Way of Pushing DATS to McAfee Clients

This batch file is shown only as an example of the method some NetAdmins use to update their client DATS.

```
@echo off
:: BadIdea.bat
::
:: This method of updating DATS is NOT RECOMMENDED. It's just an example.
copy \\myserver\virusdefs\*.dat "c:\program files\common files\network associates\virusscan
engine\4.0.xx\"
```

This script is a bad idea as it re-copies the files every time a user logs on to a computer. If a user re-logs on three times in a day, the file transfer will occur that many times as well. If NetAdmins use this logon script in a network with very many logons, the bandwidth police and performance vigilantes will call for their heads. There are better ways of getting the files pushed.

Sample script #5 – A Better Way of Pushing DATS to Clients

A Logon Script should check the client computer before copying files. In this sample, it runs a quick comparison between the server and client computer. It then copies the files only if necessary.

```
@echo off
:: ChekSame.bat
::
:: Compares a small file on the client to the same file on the server.
:: If they are different, update the AV DAT files.
::
:: Check for Norton executable. If found, jump to Norton label.
if exist "c:\program files\norton antivirus\navapw32.exe" goto norton
::
:: Check for McAfee executable. If not found, there's no AV software installed, jump to AllDone
label.
if not exist "c:\program files\network associates\virusscan\avsynmgr.exe" goto alldone
::
:: McAfee Update Section
::
:: Compare the Update.ini file on the server to the Update.ini on the client computer.
fc /l \\myserver\virusdefs\update.ini "c:\program files\common files\network associates\virusscan
engine\4.0.xx\update.ini"  | find /i "FC: no differences encountered" > nul
::
:: If the files are the same, go to "alldone" label. Otherwise, go to MInstall label.
if errorlevel==1 goto minstall
if not errorlevel==1 goto alldone
::
:: MInstall label. If above errorlevel is 1 or higher, script will jump to here.
:minstall
::
:: Install McAfee DAT files.
\\myserver\virusdefs\xdat.exe /SILENT
::
:: Copy the new update.ini from the server to the client computer.
copy \\myserver\virusdefs\update.ini "c:\program files\common files\network associates\virusscan
engine\4.0.xx\"
::
:: McAfee DAT files are now up to date. Jump to AllDone label.
goto alldone
::
:: Norton Update Section.
::
:: Compare the "definfo.dat" file on the server to the "definfo.dat" on the client computer
fc /l \\myserver\virusdefs\definfo.dat "c:\program files\common files\symantec
shared\virusdefs\definfo.dat"  | find /i "FC: no differences encountered" > nul
::
:: If the files are the same, jump to AllDone label. Otherwise, jump to NInstall label.
if errorlevel==1 goto ninstall
if not errorlevel==1 goto alldone
::
:: NInstall label. If above errorlevel is 1 or higher, script will jump to here.
:ninstall
::
:: Install Norton DAT files.
\\myserver\virusdefs\symcdefsx86.exe /q
::
:: Norton DAT files are now up to date. Jump to AllDone label.
goto alldone
::
```

```
:: AllDone label.
If AV software is missing, or if compared files are the same, or after the update completes, the
script jumps to here.
:alldone
```

In order to save bandwidth, this batch file compares a small file on the server to one on the client. If it's not the same, the script will run an executable to update the DATS.

The local server in this script is named *myserver*, with a share named *virusdefs*. This batch file requires several files to be copied to the local server share. McAfee's *4226xdat.exe* (or newer) and *update.ini* files are available at their DAT update site.[22] To simplify the script, McAfee's numbered *4226xdat.exe* (or newer) is renamed to XDAT.EXE on the local server. Norton's *symcdefsx86.exe* is available at their FTP update site.[23]

The script begins by using IF EXIST to check if a Norton executable is present. If it is, the script skips past the McAfee test to the NORTON label.

Next it uses IF NOT EXIST to check if a McAfee executable is present. If not, it will skip to the ALLDONE label, as there's nothing further the script needs to do.

Then the script runs FC (file compare) to run the comparison. It uses the /L switch to perform an ASCII comparison, which runs faster than a bit-by-bit binary comparison.

It pipes (|) the output through FIND, which includes a /I (ignore case) switch. FIND will look for the text string of "FC: No differences encountered." The results of the search are redirected to nowhere (> nul).

Note: The words beginning with "fc /l \\myserver\virusdefs\" and ending with "FC: no differences encountered" > nul" are all one command line. Your browser may auto-wrap the single line into two apparent lines.

The script will then use the FIND errorlevel. If the errorlevel is 1, the script will proceed to the MINSTALL (or NINSTALL) label. If not, if the errorlevel is 0 and it will skip to the ALLDONE label.

Finally, the script runs the DAT update utility. If the client software is McAfee VirusScan, it will also copy *update.ini*. Since this file isn't installed by default on McAfee clients, *xdat.exe* will automatically execute the first time the script is run, even if the files are already current.

Another way of comparing files between the server and client computers is to compare the file dates to each other or to the current date. A third-party program would be needed, such as Stephen Ferg's FDATE utility.[24]

When Bad Things Happen to Good People

Sadly, “the best laid schemes o’ mice and men; Gang aft a-gley.” (Burns, Robert. To a Mouse. 1785.) In the same way that disinfectants fail to kill every germ, virus infections sometimes occur despite our best-laid schemes. Before one is tempted to have his logon script include “if exist c:\windows @echo Invalid Operating System, Buy a Macintosh,” he should consider how the following batch files may help.

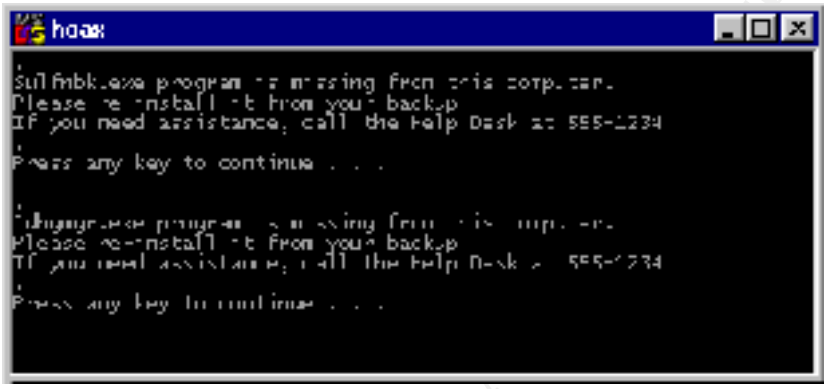
Sample Script #6 – Check if User Deleted Sulfnbk or Jdbgmgr & Notify

Sometimes a NetAdmin may need to check for missing file sets. In this example a script checks for users who may have succumbed to the SULFNBK or JDBGMGR hoaxes.[25] These hoaxes are a form of social engineering, conning users into deleting files on their own computers. Or, alternatively, they’ve converted a user into a virus. This script gently notifies the user that he deleted the file(s).

```
@echo off
:: Hoax.bat
::
:: Check to see if certain files are missing on client computer.
:: If so, notify user in logon script window.
::
:: Checks for Sulfnbk. If it exists, jump to sulfok label.
if exist c:\windows\command\sulfnbk.exe goto sulfok
::
:: If file is missing, echo instructions to console.
@echo .
@echo Sulfnbk.exe program is missing from this computer.
@echo Please re-install it from your backup
@echo If you need assistance, call the Help Desk at 555-1234
@echo .
::
:: Wait for user to "Press any key to continue."
pause
::
:: Sulfok label – if file exists, script jumps to this label.
:sulfok
::
:: Check for Jdbgmgr. If it exists, jump to jdgbok label.
if exist c:\windows\system\jdbgmgr.exe goto jdgbok
::
:: If file is missing, echo instructions to screen.
@echo .
@echo Jdbgmgr.exe program is missing from this computer.
@echo Please re-install it from your backup
@echo If you need assistance, call the Help Desk at 555-1234
@echo .
::
```

```
:: Wait for user to "Press any key to continue."
pause
::
:: Jdgbok label. If file exists, script jumps to this label.
:jdgbok
```

When the user logs in, he'll see a message like this. In this case, the user had deleted both of the files:



A similar script could check if programs or files required by policy have been deleted so the NetAdmin can respond appropriately.

Sample Script #7 – Check for HAPPY99 Virus & Clean

Happy99 is oldest of the current generation of email borne infections. It's used here as an example of a virus which creates a specific file on the client computer.

```
@echo off
:: Happy99.bat
::
:: Check for a file specific to the Happy99.Worm (W32/SKA) virus.
:: If it exists, clean the virus and notify the user in logon script window.
::
:: Check for Happy99-created file. If it doesn't exist, jump to NoHappy label.
if not exist c:\windows\system\ska.exe goto nohappy
::
:: Run "FixHappy" from server.
\\myserver\virusdefs\fixhappy.exe /auto /noreboot
::
:: If Fixhappy.exe cleaned the virus, jump to Notify label.
:: Otherwise, jump to NoHappy label.
if not errorlevel ==2 goto nohappy
if errorlevel == 2 goto notify
::
:: Notify Label. If Happy99 is found and successfully cleaned, script jumps to here.
:notify
::
```

```
:: Echo instructions to console and wait for user to "Press any key to continue."
@echo Happy99 was found on the system and removed successfully (pending reboot)
pause
::
:: NoHappy label – if Happy99 doesn't exist on client computer,
:: or was not successfully cleaned, script jumps to here.
:nohappy
```

When a virus creates a specific file or file set on a computer, IF EXIST and/or IF NOT EXIST can work simply and effectively. As this script is similar to Sample Script #6 the syntax notes aren't repeated.

This sample script uses Symantec's "FixHappy" virus removal tool.[26] The tool is manually installed in this sample on the local server named *myserver* in the *virusdefs* share. FIXHAPPY.EXE generates an errorlevel of 2 when it successfully removes the infection. The client will see a notification in their logon box that it succeeded.

Virus removal tools that are run from a logon script will work for some viruses, but not for others. For a given virus, a similar script would need to be thoroughly tested before implementation.

Sample Script #8 – Use RegEdit to Check for Klez Virus & Send Message

The most recent virus spate is sneaky. Modern viruses often don't create write specific-named files. But they normally create other markers, such as open ports or specific registry entries. In such a case, a logon batch file can find the signs of infection there. This sample batch file finds the Klez virus and sends a PopUp message to a NetAdmin.

```
@echo off
:: FindKlez.bat
::
:: Check for Kez virus startup command in the Windows registry.
:: If found, send a PopUp message to the NetAdminPC computer.
::
:: Run Registry editor and export the Run key to a temporary file named reg.txt.
start /w regedit /e reg.txt
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
::
:: Extract a specific registry value from reg.txt and send it to a temporary file.
type reg.txt | find "Krn132" > }1{.bat
::
:: Create a file fragment to begin setting temporary environment variable.
echo set Krn132=%%1>Krn132.bat
::
:: Finish setting temporary environment variable.
call }1{.bat
::
```

```
:: If environment variable was created, send PopUp message.
if %Krn132% == "C:\\WINDOWS\\SYSTEM\\krn132.exe" net send netadminpc
%computername% is infected with Klez
::
:: Cleanup - Remove temporary files and environment variable.
del reg.txt
del }?{.bat
del Krn132.bat
set Krn132=
```

This batch file finds a very precise startup command in the Windows Registry. If the Registry's Run key contains the specific entry created by Klez, "Krnl32=C:\WINDOWS\SYSTEM\krn132.exe", it will send a Windows Messenger message.

The script begins by exporting the Windows Registry "Run" key to a temporary text file. See the comments from Sample Script #2 for info on the first line's syntax.

Then the output is reduced to a single line. Reg.txt is "typed" through FIND, which searches for a line beginning with "Krn132." The output from FIND is redirected (>) to a temporary batch file.

If the computer has the Klez registry key, the temporary }1{.BAT file will contain "Krn132=C:\\WINDOWS\\SYSTEM\\krn132.exe" Note the double backslashes (\\). This is by design. The registry value itself uses just a single backslash (\), but in the exported data it's automatically doubled.

Next, the batch script takes several steps to create a temporary environment variable. To do so, it first creates a temporary file fragment. In this case, a file fragment is not as ugly as it sounds, as it's a very temporary condition. It's used here to allow the contents of two files to combine into one command.

Normally, all files end with an "End-Of-File" (EOF) marker. In this script, the EOF isn't written until another batch file is "called."

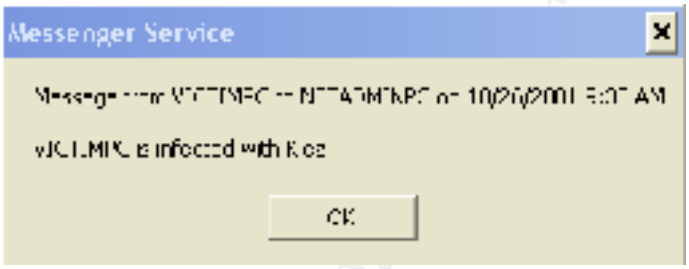
The script will use the ECHO and SET commands with the value of KRN132=%%1. The doubled percent number 1 (%%1) is a "replaceable variable," which works with input data from a source outside the command itself. In most batch files, the "%%1" input data is entered at the keyboard when the batch file is invoked. Here, "%%1" is used in a different fashion.

The output is redirected to yet another temporary batch file, KRN132.BAT. However, the "%%1" condition hasn't been fulfilled yet, so KRN132.BAT remains an incomplete fragment.

That condition is fulfilled in the next line, when the previously created }1{.BAT file is called. CALL is a command that runs another batch file without exiting the one currently running. When CALL runs, }1{.BAT will combine with the KRN132.BAT fragment to create the temporary environment variable. In the Klez test, the environment variable is "Krn132=C:\\WINDOWS\\SYSTEM\\krn132.exe." At the same time, the EOF marker is finally written.

Finally, the script checks to see if the new environment variable was indeed created. If it exists, it will use NET SEND to let the NetAdmin know that it is infected.

In this example, the infected computer named "VICTIMPC" sent a message to the computer named "NETADMINPC."



To receive the message, the *netadminpc* computer must have Windows Messenger Service running if it is a Windows NT/2000/XP system, or WinPopUp running if it's a Windows 9x system. An "Application Popup" entry will also be written to the NT/2000/XP Event Log.

Since NET SEND uses NetBIOS over TCP/IP, the *victimpc* computer must either be on the same subnet as *netadmin*, or be able to resolve the remote computer name via LMHOSTS or WINS. NET SEND can also be configured to send a message to a list of users.[27]

The pop-up method of notification works well when a NetAdmin must respond quickly to a virus infection.

Sample Script # 9 – Use NetStat to Find Bugbear Virus & Send Message

New virus attack vectors occur regularly. However, there should always be a way for a logon script to find them. This final sample logon batch file sample finds W32.Bugbear@MM,[28] a virus newly discovered on September 30, 2002.

```
@echo off
:: Bugbear.bat
::
:: Check for W32.Bugbear@MM worm by finding if Trojan port 36974 is active.
:: If found, send a PopUp message to the NetAdminPC computer.
```

```
::
:: Run Netstat.exe and redirect results to a temporary file named ports.txt.
netstat –a > ports.txt
::
:: Extract the value of 36794 from ports.txt.
type ports.txt | find "36794" > nul
::
:: If 36794 is found, jump to "Bugb" label.
:: Otherwise, jump to "NoBugb" label.
if errorlevel == 1 goto nobugb
if not errorlevel ==1 goto bugb
::
:: Bugb label. If client is infected, script jumps to here.
:bugb
::
:: If Bugbear Trojan port is active, send PopUp message.
net send netadminpc %computername% is infected with Bugbear
::
:: NoBugb Label – If BugBear port is not open, script jumps to here.
:nobugb
::
:: Cleanup - Remove temporary file.
del ports.txt
```

Most of the syntax in this script has been previously explained. The major difference in this particular script is that it uses a standard Windows program, NETSTAT.EXE. Netstat displays protocol statistics and current TCP/IP network connections. In this case, the –A switch tells NETSTAT to display all connections and listening ports. Since Bugbear creates a listening TCP Trojan port of 36794, this script will find it and message the NetAdmin.

Summary

Network Logon Scripts alone won't protect a network from Viruses, Trojans, Worms or Hoaxes. But when used in conjunction with other Anti-Virus tools, they are very useful. They can check client computers for Anti-Virus files and/or startup commands, and install Anti-Virus software as needed. Scripts can see if the Anti-Virus data files are current, updating them if necessary. If virus infections or hoaxes should occur, Network Logon Scripts can locate the infected computers. They do this by searching for files that have been added or deleted, or for virus-generated startup commands or open ports.

Network Logon Scripts can notify the user or the Network Administrator via text in the Logon Script itself, or written into files on a server. They can be output in multiple ASCII formats, such as CSV files that can be read in Microsoft Excel. Logon Scripts can also send an instant PopUp message to a computer, user, or group of users. In some cases, Logon Scripts can automatically clean the infection from the client computer.

When a Network Administrator creates a Network Logon Script, it must be tested thoroughly before deployment. This point is important enough; we'll speak from the "Department of Redundancy Department" – **Always test first, before deployment.** To do otherwise would be to invite disaster. Scripts that run fine on Windows 9x systems may fail on Windows NT and later. Yet other scripts may work well on NT but fail in Windows 9x. If there are problems with a batch file, it can be debugged in a Command-Prompt (DOS) box by typing "COMMAND /Y /C *FILENAME.BAT,*"[29] which will cause the batch script to walk step-by-step.

In their ever-vigilant task of protecting their networks, Network Administrators can utilize Network Logon Scripts as a useful tool in their Anti-Virus toolbox.

List of References

[1] Barritt, Brian & Fitzgerald, Mark. "Is It a Virus or What?" 12 Jul. 2000. Thinkquest.org. URL: http://library.thinkquest.org/C005965F/viralinfo/Virus_or_What.htm (01 Oct. 2002).

[2] Beaver, CISSP, Kevin. "Firewall best practices." 11 Jul. 2002. searchSecurity Tips & Newsletters. URL: http://searchsecurity.techtarget.com/tip/0,289483,sid14_gci838215,00.html (01 Oct. 2002).

[3] Caswell, Brian & Roesch, Marty. "SID 795." 01 Oct. 2002. Snort Signature Database. URL: http://www.snort.org/snort-db/sid.html?sid=795 (01 Oct. 2002).

[4] Evans, Billy. "Peer-Peer Networking." 29 Oct. 2001. Sans Information Security Reading Room. URL: http://rr.sans.org/threats/peer2.php (01 Oct. 2002).

[5] Ballard, Josh. "File Sharing Programs/Technologies." 26 May 2001. oofle.com. URL: http://www.oofle.com/filesharing/index.htm (30 Sep 2002).

[6] PentaSafe VigilEnt Security Management Systems. "Does Your Organization Need Information Security Policies?" PentaSafe Security Technologies, Inc. URL: http://www.pentasafe.com/publications/doyouneed.asp (01 Oct. 2002).

[7] Colburn, William. "Antivirus – a Sendmail milter." 23 Apr. 2002. New Mexico Tech. URL: http://www.nmt.edu/~wcolburn/antivirus/ (01 Oct. 2002).

[8] Microsoft Corp. "Partners, *Antivirus*." May 2001. Microsoft Exchange Server. URL: http://www.microsoft.com/exchange/partners/antivirus.asp (30 Sep 2002).

[9] Kaven, Oliver. “Anti-Virus Scanning and SMTP Throughput.” 1 Jul. 2002. Plug-In Protection, PC Magazine. URL: http://www.pcmag.com/article2/0,4149,135965,00.asp (01 Oct. 2002).

[10] Microsoft Knowledge Base Article Q49500. “List of Antivirus Software Vendors.” 06 Aug. 2002. Microsoft Product Support Services. URL: http://support.microsoft.com/default.aspx?scid=KB;EN-US;Q49500& (01 Oct. 2002).

[11] Robbins, Carol. “Kez.I – 7.2% of All Computers Infected!” 26 Apr. 2002. Panda Software. URL: http://www.pandasecurity.com/pressrelease04-26-02.htm (01 Oct. 2002).

[12] McAfee Security. “ePolicy Orchestrator.” Network Associates Technology, Inc. URL: http://www.mcafeeb2b.com/products/epolicy/default-management-solution.asp (01 Oct. 2002).

[13] Lynxwiler, Rodney. “Implementing A Norton AntiVirus Managed Infrastructure.” 21 Mar. 2002. SANS Information Security Reading Room. URL: http://rr.sans.org/malicious/norton.php (01 Oct. 2002).

[14] Schultze, F.P. “Top: Computers: Software: Operating Systems: X86: DOS: Programming Languages: Batch.” 09 Jun. 2002. dmoz Open Directory Project. URL: http://dmoz.org/Computers/Software/Operating_Systems/x86/DOS/Programming/Languages/Batch/ (01 Oct. 2002).

[15] Salmi, Timo. “/pc/*batchutil:* MS-DOS batch programming.” University of Vaasa. URL: http://garbo.uwasa.fi/pc/batchutil.html (01 Oct. 2002).

[16] Wiering, Henri. “KiXtart script.” 26 Aug. 2002. KiXtart. URL: http://kixtart.org (01 Oct. 2002).

[17] ScriptLogic Point & Click Network Administration. “Welcome to ScriptLogic.” 01 Oct. 2002. ScriptLogic Corporation. URL: http://www.scriptlogic.com (01 Oct. 2002).

[18] Castelli, Jacqueline. “Choosing Your Anti-virus Software.” 02 Apr. 2002. Sans Information Security Reading Room. URL: http://rr.sans.org/software/anti-virus.php. (01 Oct. 2002).

[19] Phelps, Eric. “Registry Operations.” Eric’s Webspace. URL: http://ericphelps.com/batch/registry/ (01 Oct. 2002).

[20] Lavedas, Thomas G. "Using Times and Dates in Batch Procedures," Method 3. Tom Ladevas' Batch File Applications. URL: http://www.pressroom.com/~tglbatch/timedate.htm (01 Oct. 2002).

[21] McAfee VirusScan. "McAfee VirusScan Administrator Manual ver. 4.5," page 58. Mar. 2002. Network Associates Technology, Inc. URL: http://download.nai.com/products/manuals/total_virus_defense/_english_us/virusscan/mcafee_versions/virusscan_multiplatform/sp1/vsc45wag.pdf (01 Oct. 2002).

[22] Network Associates. "DAT Files." 30 Sep. 2002. Network Associates Technology, Inc. URL: http://www.nai.com/naicommon/download/dats/mcafee_4x.asp (01 Oct. 2002).

[23] Symantec. "Symantec FTP Download Site." 30 Sep. 2002. Symantec Corporation. URL: ftp://ftp.symantec.com/public/english_us_canada/antivirus_definitions/norton_antivirus/static/ (01 Oct. 2002).

[24] Ferg, Stephen. "Fdate." 03 Feb. 2003. Stephen Ferg. URL: http://home.att.net/~stephen_ferg/fdate/index.html (01 Oct. 2002).

[25] Vmyths.com. "Truth About Computer Security Hysteria." 27 Sep. 2002. Rhode Island Soft Systems, Inc. URL: http://vmyths.com/ (01 Oct. 2002).

[26] Symantec Anti-Virus Research Center. "Fix Happy99.Worm Tool (FIXHAPPY.EXE)." Symantec Corporation. URL: http://securityresponse.symantec.com/avcenter/venc/data/fix.happy99.worm.html (01 Oct. 2002)

[27] Websidestory, Inc. "JSI Tip 0660 >> How do I send a message to a list of users?" 18 Aug. 1998. JSI, Inc. URL: http://www.jsifaq.com/SUBB/tip0600/rh0660.htm (01 Oct. 2002).

[28] Liu, Yana. "W32.Bugbear@mm." 30 Sep. 2002. Symantec Corporation. URL: http://securityresponse.symantec.com/avcenter/venc/data/w32.bugbear@mm.html (01 Oct. 2002).

[29] Giggleman, M.L. "About Batch Files." Jul. 1977. HAL-PC User Group. URL: http://www.hal-pc.org/journal/july97/07batchfiles.html (01 Oct. 2002).