Author: Jim Stansbury
GIAC Security Essential Certification (GSEC)
Practical Assignment 1.4b (amended August 29, 2002)
Option 1 – Research on Topics in Information Security

Archiving Event Logs

SUMMARY

Archived event logs often play an important role in the detection, investigation, and prosecution of a computer crime or other computer misuse. Configuring network devices and computer systems to produce event logs that contain valuable information is the first step in detecting computer crime. Proper archiving will further aid in the investigation and allow the event logs to be entered as evidence in court.

INTRODUCTION

The objective of this paper is to explain why it is necessary to archive, or save, event logs and to provide guidance about archiving event logs. In the context of this discussion, event logs are generally computer-generated records of a system's internal activity. The systems generating the event logs can be network devices, such as routers and firewalls, and computer systems using various operating systems, such as UNIX, Linux, and Microsoft Windows. The event logs from these network devices and computer systems can be very detailed and sometimes cryptic to the human reader. For example, they may contain a record of each packet a firewall either allowed to pass or dropped and detailed records of an operating system's internal processes. The logs can also indicate who logged on and off a system and what system resources a user accessed.

The American Heritage Dictionary defines the noun "archive" as "a place or collection containing records, documents, or other materials of historical interest." The Computer and Internet Dictionary defines the verb as "to copy files onto a tape or disk for long-term storage." Therefore, archiving system event logs is the act of copying event log files from the originating system or location to a separate place for long-term storage. Only after the system event logs are copied to a separate physical location do they become archived event logs. Just because the system event logs are configured to hold 90 days worth of events, the 90 days worth of events cannot be considered archived until they are copied off the system by a tape backup or a separate archive process.

Archiving event logs is usually performed on a schedule, such as daily or weekly. However, archiving can be performed on demand to create a snapshot of the system's activities during an incident handling procedure. This snapshot will contain events from the moment the copy was made to the beginning of the log, which may be a fixed number of days or a fixed log file size, depending on the system.

So why archive event logs at all? The main reason is to preserve evidence of malicious activity or computer crime. Archiving system event logs by itself does not add to the general security of a network or computer system. It only preserves the evidence of the system's processes and activities. But in doing so, the archive can provide evidence of computer crime, such as when a "hacker" accesses the network or disrupts the operation of a computer system.

System event logs have traditionally been used by system administrators as troubleshooting and diagnostic tools. They are still very valuable to administrators in that role today. Special system event logs are often used as activity records for accounting and can be the basis for billing customers for system usage. However, in today's world of computer crime, event log handling should be an important component in a company's security and incident handling policies. As a final step in incident handling, prosecution should be pursued. After all, the fear of prosecution is an effective deterrent, and deterrence is a component of a good corporate security policy.

CHARACTERISTICS OF GOOD EVENT LOGS

The objective of archiving is to save worthwhile, or useful, event logs, such as those having data that are valuable in the prosecution of an intruder. The value of the system's event logs to provide evidence about an intruder is directly related to what system events are written into its event log. Log files from a system that is configured for minimum event logging will probably not provide many clues or evidence of an incident. Most network devices and computer systems have the ability to granularly configure which internal process and events to write to the log file, where the log file is located, and sometimes what format the log file is written in. However, companies should not rely on the vender's default event log configuration to provide the event logging level needed to track unauthorized access. Event logging is often resource intensive on a system's performance. So, in an effort to provide as much performance as possible out of the box, a vender's default configuration may provide only minimal event logging or none at all.

Good event logs have two main characteristics: a synchronized time stamp for each event and sufficient logging level activity to produce detailed events of system activity. Each of these characteristics is discussed in detail in the following sections using the Microsoft Windows operating system as an example.

Time Synchronization

All network devices and computer systems producing event logs should be time synchronized with a time server. This is especially helpful when tracking an incident using the event logs from several different network devices and several computer systems. For example, when systems are time synchronized, it is easy to correlate suspicious repeated failed login events with the firewall event logs that show the source of the packets allowed to the pass to the system at the exact time of the failed logins. Trying to correlate time-stamped events in system logs across multiple systems whose internal clocks are not synchronized can be very frustrating and lead to flawed conclusions ("Manage Logging and Other Data Collections").

In a Windows Active Directory environment, time synchronization between Windows 2000 servers and workstations is automatic. Microsoft Windows 2000 w32Time service uses Simple Network Time Protocol (SNTP) specified in RFC 1769 for time synchronizing to support the Kerberos V5 authentication protocols. Microsoft Windows 2000 member servers synchronize time to the domain controller (The Windows Time Service).

Microsoft Windows 2000 domain controllers synchronize their times to the PDC emulator within the domain. In a forest with multiple domains, the domain controllers synchronize with other domain controllers following an Active Directory domain hierarchy-based synchronization. The PDC emulator in the forest root domain is considered the time authority for the forest and should be synchronized with an external NTP time server. A listing of public NTP time servers which can be used as an external NTP time source can be found at www.ntp.org (The Windows Time Service).

By default, Microsoft Windows NT does not have the capability to synchronize time with a time source. However, Microsoft offers the same w32Time service used in Windows 2000 for Windows NT. The w32Time service replaces the old TimeServ utility in the Microsoft Windows NT Resource Kit. Running the w32Time service on a Windows NT server will allow it to time synchronize to a Windows 2000 domain controller, a Windows NT domain controller, or an external NTP time source (W32Time Network Time Service for Windows NT 4.0). Also, a Windows 2000 server can be configured to synchronize time to a Windows NT domain controller running the w32Time service (Knowledge Base 258059).

In a Windows 2000 Active Directory forest, the PDC emulator in the forest's root domain should be synchronized to an authoritative NTP time source. In a Windows NT domain, each Windows NT domain controller should be configured to synchronize to the PDC, which in turn should be configured to synchronize to an external time source.

The clock on a Windows 2000 server ticks approximately every 10 milliseconds, which means the clock has a minimum accuracy of 10 milliseconds. However, in a Windows 2000 enterprise the SNTP protocol ensures the clocks are synchronized within 20 seconds and the clocks within a site are synchronized within two seconds (The Windows Time Service).

In summary, synchronizing time between all network devices and computer systems in the network to a NTP time source will help provide accurately time-stamped events.

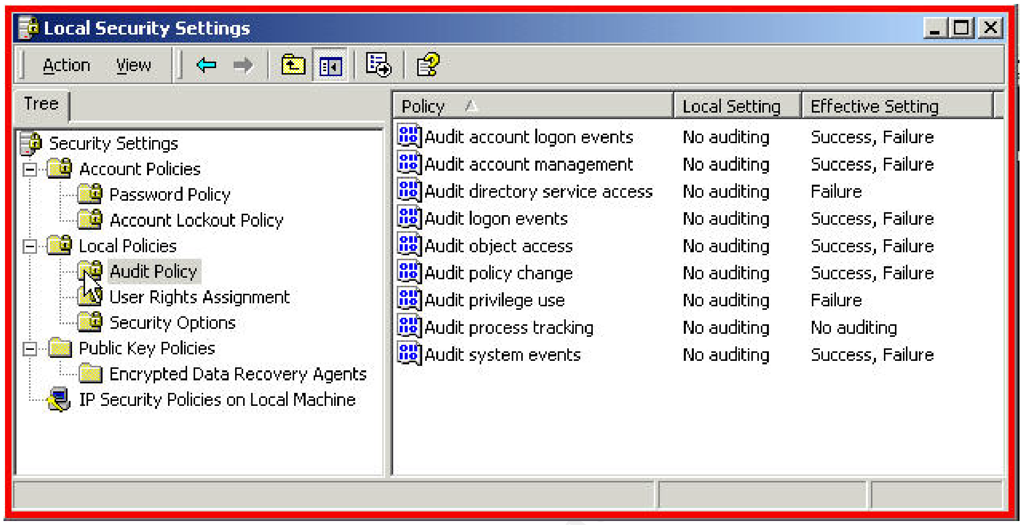
Event Log Data

As stated earlier, a system's default configuration may provide minimal to no system or security event logging. This is the case with Windows NT and Windows 2000, which do not have any event logging or auditing capabilities enabled.

In both Windows NT and Windows 2000, event logging is configured by the audit policy of each server. Additional auditing capabilities must be configured in the audit policy beyond the default settings for Windows NT and Windows 2000 servers to log system and security events that are valuable for basic forensic analysis. Most host-based intrusion detection systems and security-monitoring systems rely on an active event logging capability of the system. For example, NetIQ Security Manager requires each monitored system to have a minimal audit policy configured (Security Manager 3.5 User Guide 130).

In a Microsoft Windows environment, an audit policy can be defined as part of an overall security configuration template that is used to deploy a standard security configuration to all Windows servers across an organization. Security configuration templates configure a broad range of security settings for a server. Therefore, different security configuration templates are developed and applied to servers based on their roles in the network environment. For example, domain controllers use a security configuration template different from the general file and print servers. However, the audit policy should be standardized for all security configuration templates to provide rich and valuable event logging data across all servers.

The audit policy in Windows NT and 2000 determines how auditing is performed on the servers. There are several categories of auditing for security events, and each category can be audited separately for success or failure. The screen capture below illustrates an audit policy using the Local Security Policy console of a Windows 2000 server, which is a member of an Active Directory domain.



Policy	Local Setting	Effective Setting
Audit account logon events	No auditing	Success, Failure
Audit account management	No auditing	Success, Failure
Audit directory service access	No auditing	Failure
Audit logon events	No auditing	Success, Failure
Audit object access	No auditing	Success, Failure
Audit policy change	No auditing	Success, Failure
Audit privilege use	No auditing	Failure
Audit process tracking	No auditing	No auditing
Audit system events	No auditing	Success, Failure

Figure 1 Local Security Setting of Windows NT

The audit policy illustrated above was applied by an Active Directory group policy as a baseline audit policy to all domain controllers and member servers in the domain. In the right frame, the Effective Setting column represents the audit policy setting applied by the group policy, and the Local Setting column is the Windows 2000 default audit policy setting, which is "no auditing." The audit policy setting illustrated above is the recommended baseline audit policy referenced in Microsoft's Security Operations Guide for Windows 2000 Server and will provide a high level of auditing capability suitable for most network environments (55).

In a Windows environment, configuring the audit policy is a separate process from configuring the event log properties. The events generated by the audit policy are stored in the Windows event logs. Windows NT and 2000 servers have three event logs in common—application, system, and security. Windows 2000 servers have additional event logs, namely DNS, Directory Services, and File Replications Service. Each Windows event log has configuration options for the maximum size of the event log file and a choice of three retention methods:

- Overwrite event as needed.
- Overwrite events older than X days.
- Do not overwrite events (clear log manually).

Microsoft's Security Operations Guide for Windows 2000 Server recommends setting the maximum size of the log files to 10 megabytes (MB) and not to overwrite events. Following this recommendation will require manually clearing the event log. Manually clearing event logs may seem time-consuming at first, but it can be easily scripted or accomplished with commercially available products. If an organization is using a monitoring and archiving event log product that extracts and forwards each event to a repository database, then the event logs can be set to overwrite as needed (54).

Active Directory networks use group policies to deploy and enforce security policies across an entire enterprise very effectively. Windows NT-based domains natively cannot deploy and enforce security policies as easily. In Windows NT service pack 4, Microsoft introduced the Security Configuration Manager, which allows Windows NT servers to be configured using preconfigured security templates similar to Windows 2000 security templates. However, the Security Configuration Manager is an individual server configuration tool and does not have the capability to deploy and automatically enforce security templates to servers in a Windows NT domain (MS Security Configuration Manager for Windows NT 4).

There are commercial products available that place an active agent on each Windows NT or 2000 server. This agent is capable of installing the Security Configuration Manager and utilizing its capabilities to enforce a standard security template across a Windows NT enterprise similar to a group policy in Active Directory. One such product is NetIQ Security Manager.

Before event logs can be a valuable asset that are worth archiving, a standard audit policy and event log properties must be established for all servers in the organization and then documented in the company's computer security policy. The next step is to deploy and enforce the standards across all the servers in the enterprise. These two actions will ensure that the event logs will contain data useful for analysis and tracking incidents.

EVENT LOGS AS COURT EVIDENCE

The main purpose of archiving event logs is to provide a historical record of system activity that can be used for analysis in an investigation or evidence for prosecution in a computer crime. The following section is an overview of the laws and options with respect to entering computer records as evidence in court. The main authority governing the use of evidence is the Federal Rules of Evidence compiled by the US House of Representatives Committee on the Judiciary. Another important document on computer crime is the US Department of Justice's manual, Search and Seizing

Computers and Obtaining Electronic Evidence in Criminal Investigations, referred to as the Search and Seizure Manual.

In the past, federal courts commonly evaluated computer records as potential hearsay. Hearsay is defined as a “statement other than one made by the declarant while testifying at the trial or hearing offered as evidence to prove the truth of the matter asserted” (US House of Rep. 15). In other words, hearsay is a statement made by someone other than the witness that is offered by the witness as evidence. However, the Federal Rules of Evidence allows an exception to the hearsay rule for business records. Rule 803(6) reads in part:

> **Records of regularly conducted activity.** A memorandum, report, record, or data compilation, in any form, of acts, events, conditions, opinions, or diagnoses, made at or near the time by, or from information transmitted by, a person with knowledge, if kept in the course of a regularly conducted business activity, and if it was the regular practice of that business activity to make the memorandum, report, record or data compilation, all as shown by the testimony of the custodian or other qualified witness, or by certification that complies with Rule 902(11), Rule 902(12), or a statute permitting certification, unless the source of information or the method or circumstances of preparation indicate lack of trustworthiness. (US House of Rep.18)

This business records exception to the hearsay rule has in turn been applied to computer records. For when computer records were shown to be kept as routine business procedure, they could be classified as business records and so admitted in court as evidence (US Dept. of Justice 142).

The hearsay rule and the business records exception generally apply to human “statements”. But some types of computer records do not contain human statements at all. Therefore, as federal courts have recently become more familiar with the different types of computer records, they have made further distinctions. For evidentiary issues, computer records are classed as three different types: computer-stored records, computer-generated records, and records that are both computer-stored and computer-generated (US Dept. of Justice 143).

Computer-stored records are word processing documents, e-mail messages, or chat room messages that contain human statements. As with other evidence containing human statements, computer-stored records must comply with the hearsay rules. On the other hand, computer-generated records are produced by a computer program, are untouched by human hands, and do not contain human statements. Computer log files, telephone records, ISP dial-in records, and ATM receipts are examples.

Because computer-generated records do not contain human statements but only the output of a computer program, the evidentiary issue raised is whether the computer program that generated the record was functioning properly, which is a question of authenticity not hearsay (US Dept. of Justice 143).

Computer records that are both computer-generated and computer-stored combine the evidentiary issues of both hearsay and authenticity. An example of this type of record is a spreadsheet that not only contains direct human input but also computer calculation and output (US Dept. of Justice 143).

Establishing Authenticity

The admission of computer records generally raises two distinct issues. First, for computer-stored records containing human statements, it must be shown that the human statements are not inadmissible hearsay (US Dept. of Justice 144). This can sometimes be accomplished by proving that the computer records fall under the business records exception to the hearsay rule.

Second, the authenticity of all computer records must be established under Rule 901(a) of the Federal Rules of Evidence, which states:

> The requirement of authentication or identification as a condition precedent to admissibility is satisfied by evidence sufficient to support a finding that the matter in question is what its proponent claims. (US House of Rep. 20)

To establish the authenticity of a computer-generated record, it is not necessary for the computer programmer who programmed the computer to testify. It is only necessary for a witness to have "first-hand knowledge of the relevant facts." In one court case, computer records were authenticated by an FBI agent who was present when the files were confiscated (US Dept. of Justice 144). Under this rule, a company's computer operations staff can testify to authenticity. So to anticipate this possibility, a company needs to identify the specific personnel responsible for archiving system event logs in its security policy and standard operating procedures.

Challenges to the authenticity of computer-generated records can take two forms:

- questioning whether the records were altered, manipulated, or damaged
- challenging the reliability of the computer program

In questioning whether the computer records were altered, manipulated, or damaged, the courts have ruled in several cases that in the absence of any specific evidence, the *mere possibility* of tampering does not affect the authenticity of the computer record. United States v. Glasser, 773 F.2d 1553, 1559 (11th Cir. 1985), further stated that "the existence of an air-tight security system [to prevent tampering] is not, however, a prerequisite to the admissibility of computer printouts. If such a prerequisite did exist, it would become virtually impossible to admit computer-generated records; the party opposing admission would have to show only that a better security system was feasible" (US Dept. of Justice 145).

In challenging the reliability of the computer program, again the courts have ruled in several cases that reliability can be established by showing that the users of the program regularly depended on it, such as in the ordinary course of business. The case of United States v Salgado, 250 F.3d at 453 (6th Cir. 2001), stated that "evidence that the computer was sufficiently accurate that the company relied upon it in conducting its business" was sufficient for establishing trustworthiness. Also, the case of United States v. Moore, 923 F.2d 910, 915 (1st Cir. 1991), stated that "the ordinary business circumstances described suggest trustworthiness, ... at least where absolutely nothing in the record in any way implies the lack thereof" (US Dept. of Justice 146).

To meet such challenges to authenticity, companies need to identify the types of computer records they use regularly for security monitoring in their security policy and operating procedures. They should also document their reliance on commercial security-monitoring or host-based intrusion detection products.

Another hurdle in entering computer records as evidence in court is the best evidence rule (Rule 1002 of the Federal Rules of Evidence), which states that the "original" writing, recording, or photograph is required to prove the content. Concern is often expressed that a printout of an electronic file is not the original because the original is a collection of 1's and 0's usually stored electronically on magnetic media such as a hard disk. However, Rule 1001 of the Federal Rules of Evidence defines the "original" for electronic files as data that are "stored in a computer or similar device, any printout or other output readable by sight, shown to reflect the data accurately" (US Dept. of Justice 152).

Several court cases have upheld that an accurate printout of computer data satisfies the best evidence rule. Specifically, Laughner v. State, 769 N.E.2d 1147, 1159 (Ind. Ct. App. 2002), upheld that AOL Instant Message logs that police had cut-and-pasted into a word-processing file were considered originals (US Dept. of Justice 152). Therefore, it would follow that printouts of archived event logs would also be considered originals.

The Business Practice

Even though computer-generated records are usually not considered hearsay, it is still prudent business practice to establish that computer event logs are business records in order to use the business records exception rule if required. To qualify for the business records exception, a company must have a published computer security policy approved by the corporate officers of the company. Also, a company should document in the standard operating procedures the policies and procedures for archiving the various log files generated by the network devices and computer systems. It is also important to document a company's reliance on security-monitoring or host-based intrusion detection products so that the records from these products can be considered authentic evidence in court.

If the commercial product is one that consolidates event logs from a variety of sources and systems into a central database, it is still a good security practice to archive the original event logs in their original file format in order to satisfy the best evidence rule of requiring the "original". Also, the original event log file format will be easier for other investigative authorities to analyze rather than event logs stored in a proprietary database used by an event log monitoring product or host-based intrusion detection product.

To satisfy the federal requirement for authentication, a company should identify the personnel responsible and the procedure for archiving the event logs in the standard operating procedures document. When a situation warrants a snapshot archive of a system event log outside of the normal scheduled time for the archive, the person performing the snapshot should follow the same procedures that would be normally used during the scheduled time. In other words, the process used for the snapshot archive should be the exact process used during the scheduled archive. Only the time of the archive should be different not the process.

When a snapshot archive is made during an incident response, the person performing the archive should remain the sole custodian of the archive process and the archive itself. The archive should also be stored on write-once-read-only media, such as CD-ROM. The time and the person making the snapshot archive should be noted in the official incident response record as well.

The following are important event log archive items to include in standard operating procedures or an incident response plan:

- the personnel responsible for archiving
- the process and procedures of archiving
- the normal schedule for archiving
- the security permissions of the original event log files on the systems
- the circumstances which warrant an unscheduled snapshot archive
- storage of the archive event logs on archive media (e.g., CD-ROM, tape)

The most important thing is to document as a "standard business practice" the computer security policy, the standard operating procedures, and an incident response plan for the computer operations staff to follow. These documents and the accompanying procedures will have a direct impact on whether computer records will be allowed as evidence in the prosecution of a computer crime.

SOLUTIONS FOR ARCHIVING EVENT LOGS

There are several commercial products available for event log monitoring, host-base intrusion detection, event log analysis, and event log archiving. All of these products will enhance the system and network security when used properly.

Some of the more advanced security-monitoring products will actually forward in near real-time all events to a central database (or filter for important events) in order to correlate among all enterprise servers and provide a single event log monitoring console. This capability can help protect the events in the log from an intruder's common practice of deleting or clearing the event log or flooding it with "fake" events to cover his tracks. The events generated by the intruder's initial contact with the system would be immediately forwarded to the central event repository database for safekeeping. If the intruder cleared, deleted, or flooded the event log, this should trigger an alarm in the security-monitoring system. Two products with this capability are Microsoft Operations Manager (MOM) and NetIQ Security Manager.

It is extremely important to use commercial products that monitor or manage event logs in the computer security plan. However, these products may not provide a true archived copy of the original event log from each system. A true archived copy retains the original file format and can be stored for an indefinite period of time.

Commercial products that forward events from the host system in real-time or on a scheduled basis usually do not retain the event's original format. Instead, the information contained in the event is entered into a proprietary database that is optimized for analysis and event correlation. This is good for detecting an actual attack. But when it comes time to enter evidence in court, there is nothing like the original event log from each system to backup the evidence in the commercial product's proprietary database. Investigating authorities usually prefer copies of the event logs in the original format, not as they exist in a proprietary database.

In addition, commercial products eventually groom the events out of the database, usually based on the age of the events, in an effort to keep the database from growing too large over time. Keeping long-term archived records of the event logs requires a carefully planned database backup strategy. Retrieving event log information for an investigation requires a machine to read the backup media (tape), the database engine (e.g., Microsoft SQL Server, Oracle), and the original commercial security-monitoring software.

Companies that offer products that monitor event logs as part of a suite of capabilities include

- BindView, bv-Control, <http://www.bindview.com>
- NetIQ, NetIQ Security Manager, <http://www.netiq.com>
- Argent Software, Argent Log Consolidator, <http://www.argent.com>
- Aelita Software, Aelita InTrust, <http://www.aelita.com>
- Microsoft, Microsoft Operations Manager, <http://www.microsoft.com>

Companies that offer products that specialize in Windows event log management include

- Dorian Software, Event Archiver, <http://www.doriansoftware.com>
- TNT Software, ELM Log Manager <http://www.tntsoftware.com>
- GFI Software, GFILANguard Security Event Log Monitor <http://www.gfi.com>

A Low-Cost Solution

A low-cost solution for archiving Windows NT and 2000 event logs involves combining a few freeware utilities, the ubiquitous zip utility, and some utilities from the Windows 2000 Resource Kit with a simple Windows shell script. The example script ArchiveLogs.cmd presented here would mostly benefit a small company with a limited budget and a small number of servers.

ArchiveLogs.cmd was derived from a script by Steve Seguis published in Windows Scripting Solutions. However, ArchiveLogs.cmd offers significant changes and improvements. For example, the archived event log is more accurately time-stamped to the second, which can be useful when correlating a series of events that may span two archived event logs. Also, an MD5 message digest is made of each archived event log, and an event is recorded containing the MD5 hash in the application event log in the system from which the event log was archived.

MD5 is an algorithm used to produce a unique 128-bit hash value of an input, which is commonly a file. It is infeasible for two different files to produce the same 128-bit hash value. Just as important, an identical copy of a file will produce the same MD5 hash value as the original. An MD5 hash value of a file is often used to ensure that a file copy is identical to the original or that the file has not changed during a compression routine or during transmission (RFC1321). The ArchiveLog.cmd script produces and records an MD5 hash value for each archived event log, which can be helpful in authenticating the archived event log as evidence for court.

The ArchiveLogs.cmd Windows shell script has the following capabilities:

- Can be used by the Windows Scheduler
- Executes from a single location to archive event logs on remote servers
- Has no software installed on remote client servers
- Archives the security, system, and application event log on remote servers
- Date and time stamps (to the second) the archived event log
- Stores all archived event logs in a single location
- Uses a zip utility to compress individual archive event logs into a monthly compressed file for each server
- Optionally clears the event log after archiving
- Creates an MD5 hash of the archived event log
- Writes an event to the original server indicating the event log was archived, the name of the archived event log file, and the MD5 hash

- Uses command line options for the list of servers and type of event log to archive
- Produces a log file during execution

How to Use ArchiveLogs.cmd

ArchiveLogs.cmd is executed from a domain member server and requires domain administrator permissions. When executed by the Windows 2000 Scheduler, the script should run using a login with domain administrator permissions.

Before using ArchiveLogs.cmd, edit the script to set the following variables:

set share=	Share name on the target server
set realpath=	Directory path for the share on the target server
set tempdir=	Temporary subdirectory on the target server
set zippath=	Location of the archive event logs on the server running the ArchiveLogs.cmd

Syntax for ArchiveLogs.cmd:

ArchiveLogs.cmd *serverlist.txt app sys sec*

serverlist.txt A text file containing the list of server names by which to archive the event logs. The text file must contain one server name per line and be located in the same directory as the ArchiveLogs.cmd script. A file name must be specified.

app Designates to archive the application event log.

sys Designates to archive the system event log.

sec Designates to archive the security event log.

Example usage:

ArchiveLogs.cmd list.txt app Archives the application event log for the servers listed in list.txt file.

ArchiveLogs.cmd list.txt sec sys Archives the security and system event logs of the servers listed in list.txt file.

ArchiveLogs.cmd list.txt sys sec app	Archives the system, security, and application event logs for the servers listed in the list.txt file.

By default, the script does not clear the event log after making a copy. To clear the event log, modify the command line for elsave.exe and add –C option. Then, copy the modified script and rename ArchiveLogsClear.cmd to create two scripts, one that clears the event log and one that does not.

Because the script uses command line options to identify which servers and what type of event log to archive, the same script can be used to perform several different archive scenarios without modifying the script. For example, the same script can be used to archive

- the security event logs on domain controllers on a daily schedule
- the application event logs on all SQL Servers on a weekly schedule
- the application and security event logs on all IIS servers every six hours

The archive event logs are saved in a subdirectory specified as the environment variable “zippath” in ArchiveLogs.cmd. The NTFS security permissions on this subdirectory should be restricted to prevent unauthorized access.

The archived event logs are named using the syntax:

servername-logtype-date-time.evt

The MD5 hash of the archived event log is contained in a text file named using the syntax:

servername-logtype-date-time-MD5Hash.txt

Each event log of the same type (app, sec, sys), from the same server, and from the same month is compressed in a zip file named using the syntax:

servername-logtype-month-year.zip

The ArchiveLogs.cmd script uses the following utilities:

Vdate.exe Used to produce the data and time used in naming the archive event log file and the zipped file. This utility has extensive command line options to format the output date and time.

The example script uses the command line options vdate.exe +%b/%d/%Y/%H/%M/%S, which outputs the date and time as Mar/08/2003/17/33/48.

This format can easily be parsed in a FOR /F statement. The vdate.exe was written by David Tribble and can be found at http://david.tribble.com/programs.htm.

Elsave.exe Used to produce the event log archive. This utility makes a copy and/or clears the system, security, or application event log on a remote server. The event log is saved to a file on the server specified with the –s option.

ELSave.exe was written by Jesper Lauritsen and can be found at http://www.ibt.ku.dk/jesper/ELSave/default.htm. The executable is in the public domain, but the source code is not available.

In the example script, elslave.exe is used with the following options:

-s \\serverneme Server name of the remote server.
-F file The file name for the saved event log.
-l log The name of the event log to archive: application, system, security.
-C Clears the log. If –C is not specified the log is not cleared. ArchiveLogs.cmd does not use this option.

MD5sums.exe Used to produce an MD5 hash of the archived event log file. The MD5 message-digest algorithm was invented by Ron Rivest of RSA Data Security, Inc. and is described in RFC1321. The MD5sums.exe utility was written by Jem Berks and can be found at http://www.pc-tools.net/win32/freeware/md5sums.

Logevent.exe A Windows 2000 Server Resource Kit utility used to create an event on the target server to indicate that the event log was archived. The event description includes the file name of the archived event log and the MD5 hash of the archived event log (Windows 2000 Server Resource Kit).

Zip.exe A freeware compression utility from Info-Zip compatible to Pkzip and WinZip. Used as an alternative to commercial file compression (zip) products. The ArchiveLogs.cmd uses this utility by default (Info-Zip).

Wzzip.exe A WinZip command line optional utility used to compress the individual archive event logs into a single monthly file for each server (WinZip Command Line Support Add-On). It requires the WinZip utility. The ArchiveLogs.cmd includes an example command line syntax for this utility but is commented out.

Now.exe A Windows 2000 Server Resource Kit utility used to date and time stamp the ArchiveLogs.log file. Now.exe functions like Echo, except it produces the data and time before echoing the preceding statement (Windows 2000 Server Resource Kit).

Listing of ArchiveLogs.cmd

```
@ECHO off
setlocal
::******************************************************
:: ArchiveLogs.cmd
::******************************************************
:: This script requires the following utilities:
:: zip.exe from Info-ZIP available at www.info-zip.org OR...
:: WinZip 8.x and WinZip Command Line Support Add-On (wzzip.exe) available at
www.winzip.com
:: vdate.exe available at david.tribble.com/programs.htm
:: elsave.exe available at www.ibt.ku.dk/jesper/elsave/default.htm
:: logevent.exe available in the Windows 2000 Server Resource Kit
:: now.exe available in the Windows 2000 Server Resource Kit
::
:: ** Set these script variables for you network environment **
:: The share, realpath, and tempdir are all on the target server.
set share=c$
set realpath=c:
set tempdir=templog
::
:: Set the location of the zipped archive log files.
set zippath=c:\ArchiveLogs
:: Create the subdirectory if it does not exist.
IF NOT EXIST %zippath%\ MD %zippath%
::
:: ** End setup **
::
@ECHO Begin ArchiveLogs.cmd ***********************************>ArchiveLogs.log
:: Check command line parameters
if {%1}=={} echo missing serverlist.txt parameter & goto :EOF
if {%2}=={} echo missing parameters & goto :EOF
::
NOW Start Time>>ArchiveLogs.log
:: Use command line options 2, 3, and 4 to create loglist.txt
set target=
set ServerList=%1
set logtype2=%2
set logtype3=%3
set logtype4=%4
if defined logtype2 echo %logtype2% >loglist.txt
if defined logtype3 echo %logtype3% >>loglist.txt
if defined logtype4 echo %logtype4% >>loglist.txt
set loglist=loglist.txt
```

```
@Echo The temp directory created on the target server is:
%realpath%\%tempdir% (this will be deleted) >>ArchiveLogs.log
@ECHO The share name where the temp directory will be crerated is: %share%
(this share must exist) >>ArchiveLogs.log
@ECHO The "serverlist" environment variable is set to: %ServerList%
>>ArchiveLogs.log
@ECHO The "loglist" environment variable is set to: %loglist% >>ArchiveLogs.log
@ECHO The loglist.txt file lists the following event logs for archiving:
>>ArchiveLogs.log
type loglist.txt >>ArchiveLogs.log
:: Let the Archiving begin
FOR /F "" %%b in (%ServerList%) do call :ArchiveServer %%b
@ECHO *** End ArchiveLogs.cmd and close log file******** >>ArchiveLogs.log
NOW Stop Time>>ArchiveLogs.log
::
goto :EOF
:ArchiveServer
@ECHO ********************************************** >>ArchiveLogs.log
@ECHO Begin ArchiveServer loop to archive all event logs one server at a time
>>ArchiveLogs.log
set target=%1
@ECHO The target server during this loop is: %target% >>ArchiveLogs.log
@ECHO ******************************************************* >>ArchiveLogs.log
For /F "" %%c in (%loglist%) do call :Archive2 %%c
@ECHO ******************************************** >>ArchiveLogs.log
@ECHO End ArchiveServer loop, continue to next target server
>>ArchiveLogs.log
@ECHO ******************************************* >>ArchiveLogs.log
::
goto :EOF
::
:Archive2
@ECHO ****************************************** >>ArchiveLogs.log
@ECHO Begin Archive2 loop to archive one event log >>ArchiveLogs.log
@ECHO ***************************************** >>ArchiveLogs.log
::
set eventlog=%1
@ECHO The target server is: %target% >>ArchiveLogs.log
@ECHO The target event log is: %eventlog% >>ArchiveLogs.log
:: Create temporary directory on the target server
IF NOT EXIST \\%target%\%share%\%tempdir% MD
\\%target%\%share%\%tempdir% >>ArchiveLogs.log
IF NOT EXIST \\%target%\%share%\%tempdir% goto :Problem
::
@ECHO Set date and time variables >>ArchiveLogs.log
vdate.exe +%%b/%%d/%%Y/%%H%%M%%S >CTIME.txt
```

```
For /f "Tokens=1,2,3,4 delims=/" %%i in (CTIME.txt) do set
CTime=%%i_%%j_%%k_%%l
For /f "Tokens=1,3 delims=/" %%i in (CTIME.txt) do set zipdate=%%i_%%j
@ECHO The CTIME environment variable time stamp is: %CTIME%
>>ArchiveLogs.log
@ECHO The zipdate environment variable date stamp is: %zipdate%
>>ArchiveLogs.log
::
@ECHO Create archive of the event log >>ArchiveLogs.log
elsave -s \\%target% -F %realpath%\%tempdir%\%target%-%eventlog%-
%CTIME%.evt -l %eventlog% >>ArchiveLogs.log
::
@ECHO Make MD5 hash of archived event log and copy MD5 hash to file
>>ArchiveLogs.log
md5sums.exe -u \\%target%\%share%\%tempdir%\%target%-%eventlog%-
%CTIME%.evt > \\%target%\%share%\%tempdir%\%target%-%eventlog%-
%CTIME%-MD5hash.txt
@ECHO the MD5 hash file is: >>ArchiveLogs.log
@Type \\%target%\%share%\%tempdir%\%target%-%eventlog%-%CTIME%-
MD5hash.txt >>ArchiveLogs.log
::
@ECHO Set MD5 hash as variable >>ArchiveLogs.log
FOR /F "tokens=1,2 delims= " %%s in
(\\%target%\%share%\%tempdir%\%target%-%eventlog%-%CTIME%-
MD5hash.txt) do SET MD5=%%s %%t
@ECHO The MD5 hash environment variable is: %MD5% >>ArchiveLogs.log
::
@ECHO Create an event in the target server containing MD5 Hash of archived
event log >>ArchiveLogs.log
logevent -m \\%target% -s I -r MD5 " The MD5 hash of the archived event log for
%target% and the file name containing the MD5 hash is %MD5%"
>>ArchiveLogs.log
::
@ECHO Compress the event log into a monthly zip file >>ArchiveLogs.log
@ECHO The monthly zip file name is : %zippath%\%target%-%eventlog%-
%zipdate%.zip >>ArchiveLogs.log
zip -j -m %zippath%\%target%-%eventlog%-%zipdate%.zip
\\%target%\%share%\%tempdir%\*.* >>ArchiveLogs.log
::wzzip -m -ybc %zippath%\%target%-%eventlog%-%zipdate%.zip
\\%target%\%share%\%tempdir%\*.* >>ArchiveLogs.log
::
@ECHO ******************************************** >>ArchiveLogs.log
@ECHO End Archive2 loop, continue to archive the next event log
>>ArchiveLogs.log
@ECHO ******************************************** >>ArchiveLogs.log
::
```

```
goto :EOF
::
:Problem
@echo CAN NOT CREATE C:\%tempdir% on TARGET SERVER
>>ArchiveLogs.log
```

LIST OF REFERENCES / WORKS CITED

"Archive." Def. The American Heritage Dictionary of the English Language. 3rd ed. 1992.

"Archive." Def. Computer and Internet Dictionary. Microsoft Corporation. 1998. Portions, The Microsoft Press Computer Dictionary, 3rd ed., Microsoft Press 1997.

Berkes, Jim. MD5sums.exe MD5 Utility Ver. 1.1. <http://www.pc-tools.net/win32/freeware/md5sums/.

Info-Zip. Zip.exe Ver. 2.3. Nov. 29 1999. <http://www.info-zip.org/Zip.html>.

Knowledge Base 258059. How to Synchronize the Time on a Windows 2000-Based Computer in a Windows NT 4.0 Domain. Microsoft Corp. Oct. 2002. <http://www.microsoft.com/technet>.

Lauritsen, Jesper. ELSave Event Log Utility. Ver. 0.4. http://www.ibt.ku.dk/jesper/ELSave/default.htm

"Manage Logging and Other Data Collections." CERT Security Improvements Modules. < http://www.cert.org/security-improvement/practices/p092.html>.

MS Security Configuration Manager for Windows NT 4. Microsoft Corp. 1998. <http://www.microsoft.com/technet>.

RFC1305. Network Time Protocol (Version3). David L. Mills. March 1992 <http://rfc.net/rfc1305.html>.

RFC1321. The MD5 Message-Digest Algorithm. R. Rivest. April 1992 <http://rfc.net/rfc1321.html>.

RFC1769. Simple Network Time Protocol (SNTP). D. Mills. March 1995 <http://rfc.net/rfc1769.html>.

Security Manager 3.5 User Guide. NetIQ Corp. 15 March 2002.

Security Operations Guide for Windows 2000 Server. Microsoft Corp. <http://www.microsoft.com/technet>.

Seguis, Steve. "Event-Log Auditing, Part 1." Windows Scripting Solutions. Feb. 2003.

Tribble, David. VDate Utility. < http://david.tribble.com/programs.htm>.

United States. Dept. of Justice Computer Crime and Intellectual Property Section Criminal Division. Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations. Washington: GPO, 2002.

---. House of Representatives Committee on the Judiciary. Federal Rules of Evidence. Washington: GPO, 2001.

W32Time Network Time Service for Windows NT 4.0. Microsoft Corp. <http://www.microsoft.com/technet>.

Windows 2000 Server Resource Kit. Microsoft Press. CD-ROM. <http://www.microsoft.com/MSPress/books/4355.asp>.

The Windows Time Service. Shala Brandolini, Darin Green. Microsoft Corp. April 2001. <http://www.microsoft.com/technet>.

WinZip Command Line Support Add-On. Ver. 1.0. WinZip Computing Inc.. (http://www.winzip.com/wzcline.htm).