

## Maintaining Credible IIS Log Files

*Mark Burnett* 2002-11-13

Many network administrators by now have encountered serious Web server intrusions that have resulted in legal action. Often IIS logs are the primary evidence used to track down Web intruders. But what would happen if the credibility of your IIS logs was challenged in court? What if the defense claimed the logs were not reliable enough to be admissible as evidence?

I once investigated a serious intrusion as part of a criminal investigation. An intruder broke into an IIS server, uploaded some tools, and then accessed the company's internal database. We knew approximately when the intrusion occurred, but we did not know which of several hundred Web sites on a dozen servers was compromised.

As I mined through hundreds of log files stored on the Web servers, I came across one log file that had, among the thousands of log entries, a single blank line. I checked the last modified date of that file and found that it had been modified two days after the log file was closed. Hundreds of megabytes of log file evidence suddenly became useless due to a single blank line. Because the log files were stored on the same server that was compromised, the intruder could have easily removed evidence or, worse, replaced it with false evidence pointing to someone else. The modification of one log file is compelling reason to question the validity of every log file on that server.

Proving that your log files are credible requires that you provide convincing arguments that they are trustworthy and therefore valid as evidence. You must take measures to protect the accuracy, authenticity, and accessibility of your IIS log files. Although there are many legal complexities and you should always seek your own legal advice in these cases, below are some tips that should increase the credibility of your IIS logs.

### Log File Accuracy

Accuracy means that you can prove that your log file data truly represents the activity on your Web server. Even the smallest inaccuracy can bring into question the validity of the entire set of data. The following steps help ensure that your data is accurate:

**Log Everything** - Configure your IIS logs to record every available field. While some admins see little value in storing this extra information, every field has some significance in a forensics

investigation. I once examined some logs after an IIS intrusion and saw activity indicating the attacker had created files in the C:\WINNT directory. However, I was not able to locate those files anywhere on the hard drive. Looking more closely at the logs, I noticed that the server's name did not match the name recorded in the IIS logs. It turns out that the company recently migrated the Web site to a new server. Checking the old server, I immediately found the files in question. Without the logged host name, I would not have been able to make the connection.

Furthermore, gathering information about Web visitors helps establish that an attack came from a specific computer system or logged in user. For example, suppose a defendant claims a hacker had broken into his computer and installed a backdoor proxy server, then used that backdoor proxy to attack other systems. How do you prove that the traffic came from a specific user's Web browser or was a proxied attack from someone else? While this cannot always be proven, the more information you collect, the better chance you have of making this case.

**Keeping Time** - Synchronize your IIS servers to an external time source using the Windows Time service. If you use a domain, the time service will automatically be synchronized to the domain controller. On a standalone server, you can synchronize to an external source by setting the following registry entries:

**Key:** HKLM\SYSTEM\CurrentControlSet\Services\W32Time\Parameters\

**Setting:** Type

**Type:** REG\_SZ

**Value:** NTP

**Key:** HKLM\SYSTEM\CurrentControlSet\Services\W32Time\Parameters\

**Setting:** NtpServer

**Type:** REG\_SZ

**Value:** tock.usno.navy.mil (see <http://tycho.usno.navy.mil/ntp.html> for a list of public NTP servers.)

**Key:** HKLM\SYSTEM\CurrentControlSet\Services\W32Time\Parameters\

**Setting:** Period

**Type:** REG\_SZ

**Value:** 24 (Indicates times per day to synchronize. A value of 24 synchronizes once every hour, you may need less.)

Another time issue is that IIS records logs using UTC time. This is supposed to help synchronization issues when running servers in multiple time zones. However, Windows calculates UTC time by offsetting the value of the system clock with the system time zone. The only way to be sure the UTC time is correct is to ensure that the local time zone setting is accurate.

One trick is to verify that this is to set IIS to roll over logs using local time. You can later verify the server's time zone setting by looking at the first entries in the log file. If your server is set at UTC -0600, then the first log entries should appear around 18:00 (00:00 - 06:00 = 18:00). Because UTC doesn't follow daylight savings, you must also consider the date. For example, UTC - 6:00 will actually be -5:00 half the year.

**Use Multiple Sensors** - It is hard to disprove a log entry if two separate devices record the same information. By combining logs from several devices, you strengthen the value of each. Firewall logs, IDS logs, and even something as simple as TCPDump can help prove that an IP address hit a specific server at a specific time. See <http://www.iisecurity.net/4361.htm> for an example of using Snort to supplement IIS logs.

**Avoid Missing Logs** - One problem with IIS logs is that if the server does not get any hits in a 24-hour period, no log file is created. But when no log file exists, there is no way of knowing if the server got no hits (say it was offline for a day) or if the log file was actually deleted. To avoid this problem, I like to schedule a few hits each day to ensure there is always a log.

To do this, I use Graburl which you can download from <http://www.kiraly.com/software/utilities/graburl/>

Using the Task Scheduler, I schedule two hits to the Web server: one from localhost and the other from an external host. The command line is simple:

```
Graburl.exe www.example.com
```

The reason for scheduling two hits is that the first from localhost verifies that the server is running and the second verifies that it is visible on the Internet. Further, the second hit also verifies the time synchronization. If the second hit is scheduled to occur at 1:00 AM every day, the corresponding log entry should always occur at 1:00 AM. In general, scheduled requests help prove that the logging mechanism is functioning properly.

If the Web server is powered off for a period of more than 24 hours, no log file will be recorded, but your EventLog will indicate that the server had been powered off. Following these steps, if a log file is missing, it is probably because the file was intentionally deleted.

## Log File Authenticity

Log files can be said to be authentic if it can be proven that they have not been modified since they were originally recorded. IIS log files are simple text files that can easily be modified. The file date and time stamps can also easily be modified. In their default state, IIS log files cannot be proven authentic, but by following a few tips you can remedy this.

**Move the Logs** - To begin ensuring authenticity, move the IIS logs off the Web server. If a server has been compromised, you must consider that the log files too could have been compromised. Move the logs to a master server then move them offline to a tape, CD, or WORM device as quickly as possible.

**Signatures, Encryption & Checksums** - The only way to be absolutely sure a log file has not been modified is to sign and encrypt the logs using PGP or some other public-key encryption scheme. File signatures are helpful because if a single file is corrupted, it does not invalidate the rest of the logs. You can also use a tool such as [Fsum](#) to quickly generate MD5 hashes for the files. Store the signatures and hashes with the logs but also store a secure copy in a separate location.

Note that if you use an automated process for signing log files, you should always follow up with a manual signature by a trusted administrator.

When encrypting files, you should consider what impact that will have on the created, modified, and access dates. You may want to record these dates in a separate location by using a utility such as [Fdir](#).

**Work With Copies** - When doing any log file analysis, never work with the original files. Make copies before performing any post-processing or log file analysis. Making sure that original logs are never touched helps you establish that they are still authentic and in their original form. If you use log files as court evidence, you must present original files in their original form. Note that in the United States, the Federal Rules of Evidence state that an accurate printout can also be considered original evidence (see [Federal Rules of Evidence 1001\(3\)](#)).

**Ensure System Integrity** - You should always keep up to date on service packs and hotfixes to ensure that your system files are valid. You should also audit all changes to binary files in your WINNT directory. If an intruder is able to modify system files that record log files, the usability of the log files as evidence suddenly come into question.

**Have a Process** - Keep in mind that a well-established and documented process can actually help establish authenticity. An established procedure that produces consistent results may help establish that the files are valid and authentic. Furthermore, be sure to have a documented and consistent method for capturing additional evidence (such as using network diagnostic utilities against an attacker's IP address) because business records created in anticipation of litigation may not always be admissible in court. This is especially true if law enforcement asks you (without proper court orders) to use a tool such as a sniffer on your network to gather additional evidence.

Establishing a process means creating a document that outlines each manual or automated step taken. Furthermore, any scripts you use in log file processing should also contain comments explaining exactly what processing is taking place. The techniques you use in your process should be generally accepted procedures for log file management.

## **Access Control**

Once a log file is created, it is important to prevent the file from being accessed and audit any authorized and unauthorized access. If you properly secure and audit a log file using NTFS permissions, you will have documented evidence to help establish its credibility.

**Restrict File Access** - A log file needs certain permissions so that IIS is able to write to the file. But after the log is closed, no one should have permissions to modify the file contents. You may want to consider scheduling a command to lock down file permissions and auditing after a log file is closed. Also, when you move log files, be sure that NTFS permissions are set correctly in the new location.

**Chain of Custody** - As you move log files from the server and later to an offline device, you should keep track of where the file goes. This can be done either through technical or non-technical methods. For example, one client of mine seals their backup tapes and uses a label that can be used to record the physical movement of the tape. Tracking custody of evidence is

especially important when retrieving the contents of a backup in a criminal investigation.

Keep in mind that the everyday process of gathering logs may some day be part of the gathering of evidence in a criminal investigation. You should always treat your IIS Web server as if it is already a crime scene and handle your logs as if they are already evidence.

Although there are many laws and legal jurisdictions in the world, log files are often treated the same as business records and you must follow certain rules to ensure their admissibility or weight. There is not always a clear definition of what is admissible and what is credible. But just as with a court witness, it often comes down to believability. And the more documented evidence you have, the more believable your IIS logs will become.

Of course, your local evidence and privacy laws may vary and your lawyer may have a different opinion about what makes a log file credible. Discuss the issues with your lawyer and establish a process that will ensure the accuracy, authenticity, and controlled access of your IIS log files.

## References

[Federal Rules of Evidence](#)

[Searching and Seizing Computers and Obtaining Electronic Evidence in Criminal Investigations](#)

[Computer Records and the Federal Rules of Evidence](#)

*Mark Burnett is an independent security consultant and freelance writer who specializes in securing IIS. He is co-author of [Maximum Windows Security \(SAMS\)](#), [Special Ops: Host and Network Security for Microsoft, Unix, and Oracle \(Syngress\)](#) and contributed to [Dr. Tom Shinder's ISA Server and Beyond: Real World Security Solutions for Microsoft Enterprise Networks \(Syngress\)](#). Mark is a regular contributor to [SecurityFocus](#) as well as other security-related publications. You can contact Mark at [mb@xato.net](mailto:mb@xato.net) or visit his Web site at [www.iissecurity.net](http://www.iissecurity.net).*

[Privacy Statement](#)

Copyright 2006, SecurityFocus