Implementation

The ways to enable and configure logging will vary from one computer and software system to another, and should be covered in the system documentation. This section cannot deal with such detailed instructions, but identifies a number of common topics that have been found to be useful in many different circumstances.

Central Logging

One of the uses of logfiles is in the investigation of attacks on computer systems. However, a successful attack will often give the intruder complete control of the system, including the ability to delete or modify files. Tampering with logfiles or the programs used to access them to conceal evidence of the break-in is normally one of the intruder's first priorities. The risk of finding deleted or corrupted evidence can be greatly reduced by holding logs on a different computer. A successful intruder may still have the ability to add records to the logfile but is much less likely to be able to rewrite history. Having logs on a dedicated central system can also make it much easier to deal with logs from a large number of different computers, as these will automatically be gathered into one place.

The most commonly used system for writing remote logfiles is the syslog service, which was originally written for UNIX® but is now available for most other operating systems. Syslog allows messages of a standard form to be both written to a local file and transmitted over the network to one or more central logging hosts. Syslog sends each message separately, so there should be little delay between the local and remote copies of the logfile being updated. Messages are automatically timestamped by the service, but this still relies on the system clocks being synchronised to some common standard (see the next section).

Two potential problems need to be borne in mind when using the syslog service over a network. The first is that messages are sent over the network using the User Datagram Protocol (UDP), so there is no guarantee that they will arrive at their destination. On a Local Area Network with little congestion this is not normally a problem, but messages may be lost if there is a high traffic load on the network. Syslog over UDP may not be sufficiently reliable to be used over a Wide Area Network, especially as some types of attack may themselves increase the likelihood of congestion and loss of UDP messages. Some syslog implementations allow TCP to be used instead, to increase reliability, though this may be at the cost of longer delays between the event and its being logged. For these reasons it is good practice to store logs locally on the system that generates them, as well as sending them to a central logging host. The other problem is that use of remote logging can itself lead to network congestion if a very large number of error messages are generated. For example a denial of service attack on a network will be made much more effective if the target systems are trying to report each attack over the same network. Avoiding this problem requires careful design of the logging system. One option is to summarise batches of repeated log messages into a single message including the number of repeats in a particular period of time.

Highly reliable central logging systems can be built by using separate network connections to carry logging messages. In such a system, the critical computers generating logs will have dedicated network links to the central logging host. These links should not be used for normal traffic, nor connected to the production network. Such systems can also protect against sophisticated attacks where a denial of service attack against the logging system is used to conceal an intrusion into vital production services. The ultimate in tamper-resistance is to write logs immediately to an unerasable form of storage such as a write-once, readmany CD-R or DVD-R drive.

Timestamps

Most incidents involve more than one computer so it is common for an investigation to have to deal with logfiles from many different systems, possibly at different sites or even in different countries. Entries in logfiles that refer to the same event are most commonly matched by comparing their timestamps. To ensure that the times from different logs can be compared within a site, or with a complaint made from the other side of the world, it is essential that the times of different computers making the logs be synchronised to an international standard. The Network Time Protocol (NTP) is the common way to do this across computer networks and countries. A central NTP service is provided, linked to a number of atomic clocks keeping standard international time, which Janet sites can and should join. For more details of this service, see: Janet Network Time Service [1].

International incidents often occur across different time zones so that the numeric values of time may not be directly comparable. All systems should be set up to record the time zone against which they are logging, and whether daylight saving has been applied. Unfortunately there are a number of different formats for recording this information. Indeed some time zone names are not unique, so correlating logs is often harder than it should be. When reporting an incident, you should always include the time zone with respect to Coordinated Universal Time (UTC, for example ‘16:19:00 +0100’ for British Summer Time) and whether timestamps are synchronised to an international standard. Without this a great deal of effort can be wasted.

Automated Processing of Logs

Logfiles can grow very large, and routinely scanning them by eye may not be possible. There are a number of computer programs that can help to monitor logs and many sites have written their own scripts for this purpose. Such programs aim to identify patterns in the logfiles, and they can be very effective at identifying common, known problems. However a program is unlikely ever to be as good as a human at spotting new or unusual patterns. A compromise solution may be to use programs to filter out entries that are known to be harmless (though even these should be checked occasionally) and well-known problem patterns, and then to scan the remaining information by eye. As new patterns are identified they can be added to the known-good and known-bad filters. This process of tuning can be time-consuming, but is the most effective way to extract information from the logs. Raw logs should still be kept, subject to the issues relating to Data Retention in Section 2.2, as it can be useful to review them when new patterns are identified. It is very common for hostile activity to take some time to be noticed, and reassuring at that stage to be able to review older logs to determine when it actually began.

Graphing Activity

Humans are quite good at spotting patterns in textual information, but they are extremely good at finding them in graphical representations. A highly efficient way to monitor the health of any system is to graph some appropriate measures of its performance. Graphs of network traffic levels such as those provided by the Janet Netsight System [2] are fairly commonly used. Less frequent, but very useful, are graphs of, for example, number of requests to a web server, number of failed and successful logins to a network, or system idle time or memory usage. Regular patterns in these graphs will quickly become familiar to the operator. Once the system's normal behaviour is well known, unexpected changes will often be spotted without conscious thought. Detailed investigation can then be done using the original logfiles.

Links
[1] http://community.ja.net/library/janet-services-documentation/network-time-service
[2] http://netsight.ja.net/