Six Mistakes of Log Management

Anton Chuvakin, Ph.D., GCIA, GCIH, GCFA
Chief Logging Evangelist
LogLogic, Inc

NOTE: originally published in "CSI Journal" (special CSI issues, June 2007)

Updated August 2007

The Mistakes

#1 not logging at all.
#2 not looking at the logs
#3 storing logs for too short a time
#4 prioritizing the log records before collection
#5 ignoring the logs from applications
#6 only looking at what they know is bad

Since I wrote my log mistakes paper a few years ago, the domain of log analysis changed a lot. Many factors affected it; among those are new regulatory compliance requirements, wider adoption of "best practice" and governance frameworks such as ISO, COBIT and ITIL as well as new technologies with their log files. New standards, such as NIST 800-92 Guide [PDF], have been created.

Thus, I am updating the article with newly committed mistakes as well as new perspectives on the old ones. This article, just like its predecessor, again covers the typical mistakes organizations make while approaching management of computer logs and other records produced by IT infrastructure components.

As digital technology continues to spread ("A web-enabled fridge, anybody? Its just a few (!) grand today, you know" ☺) and computers start playing even more important role in our lives (I do have a penchant for the obvious, don't I? ☺), the records that they produce, a.k.a. logs, start to play bigger and bigger role. From firewalls and intrusion prevention systems to databases and enterprise applications to wireless access points and VOIP gateways, logs are being spewed forth at an every increasing pace. Both security and other IT components not only increase in numbers, but often come with more logging enabled out of the box. Example of that trend include Linux systems as well as web servers that now ship with increased level of logging. All those systems, both legacy and novel, are known to generate copious amounts of logs, audit trails, records and alerts, that beg for constant attention. Thus, many companies and government agencies are trying to set up repeatable log collection, centralization and analysis processes and tools.

However, when planning and implementing log collection and analysis infrastructure, the organizations often discover that they are not realizing the full promise of such a system

and, in fact, sometimes notice that the efficiency is not gained but lost as a result. This often happens due to the following common log management mistakes.

We will start from the obvious, but unfortunately all too common even in this age of Sarbanes-Oxley and PCI. This mistake destroys all possible chances of benefiting from logs.

It is the mistake **#1**: **not logging at all.** A more exciting flavor of this mistake is: "not logging and not even knowing it until it is too late."

How can it be "too late", some say? "Its just logs!" Welcome to 2007! Not having "just logs" can lead to losing your income (PCI that contain logging requirements implies that violations might lead to your credit card processing privileges being cancelled by Visa or Mastercard and thus putting you out of business), reputation (somebody stole a few credit card number from your database, but the media reported that all of the 40 million credit card have been stolen since you were unable to prove otherwise) or even your freedom (see various Sarbanes-Oxley horror stories in the media)

Even better-prepared organizations fall for this one. Here is a recent example. Does your web server have logging enabled? Sure, it is a default option on both of the popular web servers: Apache and Microsoft IIS. Does your server operating system log messages? Sure, nobody cancelled */var/log/messages*. But does your *database*? Oops! Default option in Oracle is to not do any data access audit logging. Maybe MS SQL fares better? Nope, same thing, you need to dig deep in the system to even start a moderate level of audit trail generation (see more on this in my database logging paper).

Thus, to avoid this mistake one needs to sometimes go beyond the defaults and make sure that the software and hardware deployed does have some level of logging enabled. In case of Oracle, for example, it might boil down to making sure that the '*audit_trail'* variable is set to '*db'* (as well as a few other tweaks); for other systems it might be more complicated.

#2 Not looking at the logs is the second mistake. While making sure that logs do exist and then collecting and storing them is important, it is only a means to an end – knowing what is going on in your environment and being able to respond to it as well as possibly predict what will happen later. Thus, once the technology is in place and logs are collected, there needs to be a process of ongoing monitoring and review that hooks into actions and possible escalations, if needed. In addition, personnel reviewing or monitoring logs should have enough information to be able to determine what they really mean and what – if any – action is required.

It is worthwhile to note that some organizations take a half-step in the right direction: they only review logs (provided they didn't commit the first mistake and they actually have something to review) after a major incident (be it a compromise, information leak or a mysterious server crash) and avoid ongoing monitoring and log review, often by quoting "the lack of resources". This gives them the reactive benefit of log analysis, which is

important, but fails to realize the proactive one – knowing when bad stuff is about to happen or become worse. For example, if you review logs, you might learn that the failover was activated on a firewall, and, even though the connection stayed on, the incident is certainly worth looking into. If you don't and your network connectivity goes away, you'd have to rely on your ever-helpful logs in investigation why *both* failover devices went down … In fact, looking at logs proactively helps organizations to better realize the value of their existing network, security and system infrastructure.

It is also critical to stress that some types of organizations *have to* look at log files and audit tracks due to regulatory pressure of some kind. For example, US HIPAA regulation compels medical organizations to establish audit record and analysis program (even though the enforcement action is notorious lacking). In a more extreme case, PCI (Payment Card Industry) data security standard has provisions for both log collection and log monitoring and periodic review, highlighting the fact that collection of logs does not stand on its own.

#3 The third common mistake is **storing logs for too short a time**. This makes the security or IT operations team think they have all the logs needed for monitoring and investigation or troubleshooting and then leading to the horrible realization after the incident that all logs are gone due to their shortsighted retention policy. It often happens (especially in the case of insider attacks) that the incident is discovered a long time – sometimes many months - after the crime or abuse has been committed. One might save some money on storage hardware, but lose the tenfold due to regulatory fines.

If low cost is critical, the solution is sometimes in splitting the retention in two parts: shorter-term online storage (that costs more) and long-term offline storage (that is much cheaper). A better three-tier approach is also common and resolves some of the limitations of the previous one. In this case, shorter-term online storage is complemented by a near-line storage where logs are still accessible and searchable. The oldest and the least relevant log records are offloaded to the third tier, such as tape or DVDs, where they can be stored inexpensively, but without any way to selectively access the needed logs. More specifically, one financial institution was storing logs online for 90 days, then in the near-line searchable storage for 2 years and then on tape for up to 7 years or even more.

#4 The fourth mistake is related to log record prioritization. While people need a sense of priority to better organize their log analysis efforts, the common mistake nowadays is **in prioritizing the log records before collection.** In fact, even some "best practice" documents recommend only collecting "the important stuff." But what **is** important? This is where the above guidance documents fall short by not specifying it in any useful form. While there are some approaches to the problem, all that I am aware of can lead to glaring holes in security posture or even undermine the regulatory compliance efforts.

For example, many people would claim that network intrusion detection and prevention logs are inherently more important than, say, VPN concentrator logs. Well, it might be true in the world where external threats completely dominate the insider abuse (i.e. not in this one). VPN logs, together with server and workstation logs, is what you would most likely

need to conduct an internal investigation about the information leak or even a malware infection. Thus, similar claims about the elevated importance of whatever other log type can be similarly disputed, which would lead us to a painful realization that you do need to collect **everything.** But can you? Before you answer this, try to answer whether you can make the right call on which log is more important even before seeing it and this problem will stop looking unsolvable. In fact, there are cost-effective solutions to achieve just that.

The mistake #**5** is **in ignoring the logs from applications**, by only focusing on the perimeter and internal network devices and possibly also servers, but not going "higher up the stack" to look at the application logging.

The realm of enterprise applications ranges from SAPs and PeopleSofts of the world to small homegrown applications, which nevertheless handle mission-critical processes for many enterprises. Legacy applications, running on mainframes and midrange systems, are out there as well, often running the core business processes as well. The availability and quality of logs differs wildly across the application, ranging from missing (the case for many home-grown applications) to extremely detailed and voluminous (the case for many mainframe applications). Lack of common logging standards and even of logging guidance for software developers lead to many challenges with application logs.

Despite the challenges, one needs to make sure that the application logs are collected and made available for analysis as well as for longer term-retention. This can be accomplished by configuring your log management software to collect them and by establishing a log review policy, both for the on-incident review and periodic proactive log review.

#6 Even the most advanced and mature organizations fall into the pitfall of the sixth error. It is sneaky and insidious, and can severely reduce the value of a log analysis project. It occurs when organization is **only looking at what they know is bad** in the logs**.** Indeed, a vast majority of open source and some commercial tools are set up to filter and look for bad log lines, attack signatures, critical events, etc. For example, "swatch" is a classic free log analysis tool that is powerful, but only at one thing: looking for defined bad things in log files. Moreover, when people talk about log analysis they usually mean sifting through logs looking for things of note.

However, to fully realize the value of log data one has to take it to the next level to log mining: actually discovering things of interest in log files without having any preconceived notion of 'what we need to find'. It sounds obvious - how can we be sure that we know of all the possible malicious behavior in advance – but it disregarded so often. Sometimes, it is suggested that it is simpler to just list all the known good things and then look for the rest. It sounds like a solution, but such task is not only onerous, but also thankless: it is usually even harder to list all the good things than it is to list all the bad things that might happen on a system or network. So many different things occur, malfunction or misbehave, that weeding out attack traces just by listing all the possibilities is not effective. A more intelligent approach is needed! Some of the data mining (also called "knowledge discovery in databases" or KDD) and visualization methods actually work on log data with

great success. They allow organizations to look for real anomalies in log data, beyond ‘known bad’ and ‘known good’.

To conclude, avoiding the above six mistakes will take your log management program to a next level and enhance the value of the existing security and logging infrastructures.

Dr Anton Chuvakin, GCIA, GCIH, GCFA (http://www.chuvakin.org) is a recognized security expert and book author. He currently works at LogLogic as a Chief Logging Evangelist. He was previously a Security Strategist with a security information management company. He is an author of a book "Security Warrior" and a contributor to "Know Your Enemy II", "Information Security Management Handbook", "Hacker's Challenge 3" and “PCI Compliance.” Anton also published numerous papers on a broad range of security subjects. In his spare time he maintains his security portal http://www.info-secure.org and several blogs (e.g. http://chuvakin.blogspot.com).