“Five Mistakes of Security Log Analysis”

Anton Chuvakin, Ph.D., GCIA, GCIH
Security Strategist with netForensics

The article covers the typical mistakes organizations make while approaching analyzing audit logs and other security-related records produced by security infrastructure components.

As the IT products and services market is growing, organizations are deploying more security solutions to guard against the ever-widening threat landscape. Firewalls are humming along, anti-virus solutions are deployed, intrusion detection systems are installed and new anti-spyware and web protection tools are being rolled out. All those devices are known to generate copious amounts of audit 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.

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. This happens due to the following common log analysis mistakes.

We will start from the obvious, but critical one. **Not looking at the logs** is the first mistake. While collecting and storing logs is important, it is only a means to an end – knowing what is going on in your environment and responding to it. 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.

It is worthwhile to note that some organizations take a half step in the right direction: they only review logs after a major incident. 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.

In fact, looking at logs proactively helps organizations to better realize the value of their existing security infrastructure. For example, many complain that their network intrusion detection systems (NIDS) do not give them their money’s worth. A large reason for that is that NIDS often produce false alarms of various kinds (“false positives”, etc) leading to decreased reliability of their output and inability to act on it. Comprehensive correlation of NIDS logs with other records such as firewalls logs, server audit trails as well as vulnerability and network service information about the target allows companies to “make NIDS perform” as well as gain new detection capabilities from such correlation (such as real-time blocking and attack mitigation).

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.

The second common mistake is **storing logs for too short a time**. This makes the security team think they have all the logs needed for monitoring and investigation (and save money on storage hardware at the same time) and then leading to the horrible realization after the incident that all logs are gone due to their retention policy. It often happens (especially in the case of insider attacks) that the incident is discovered a long time after the crime or abuse has been committed.

If low cost is critical, the solution is in splitting the retention in two parts: shorter-term online storage (that costs more) and long-term offline storage (that is much cheaper). For example, archiving old logs on tape allows for cost effective offline storage, while still enabling analysis in the future (after reinserting into the main collection and analysis framework). Thus, long-term storage solution does not need to be capable of advanced analysis and mining, available for short-term online analysis application. Dedicated log storage LDBMS can also be used in some case for reduced cost long-term storage.

Third mistake is **not normalizing logs.** What do we mean by “normalization” here? It means we can convert the logs into a universal format, containing all the details of the original message but also allowing us to compare and correlate very different log data sources such as UNIX Syslog and Windows event log. Across different application and security solutions, log format confusion reigns unchallenged. Some prefer SNMP, others favor classic UNIX syslog with all its limitations, and relational databases and proprietary files are also common. While some say that XML is becoming a standard way to log, it should be noted that XML is not a specific format and different devices uses wildly different XML document types (or “schemas”).

Lack of standard logging format leads to companies needing different (and expensive!) expertise to analyze the logs. Not all skilled UNIX administrators who understand syslog format will be able to make sense out of an obscure Windows event log record (and vice versa). The situation is even worse with security solutions since people commonly have experience with a limited number of commercial intrusion detection and firewall solutions and thus will be lost in the log pile spewed out by a different device type. As a result, a common format that can encompass all the possible messages from security-related devices is essential for analysis, correlation and ultimately for decision-making.

Assuming that logs are collected, stored for a sufficiently long time and normalized (and thus all of the previous 3 mistakes are safely avoided), what else lurks in the muddy sea of log analysis? Logs are there – but where do we start? Should we just go for a high-level summary, look at most recent event or what? The fourth error is **not prioritizing log records**. Some analysts tend to just “shut down” after trying to chew a ‘king-size’ chunk of log data, without getting any real sense of priority. How do eat an elephant? Methodically and piece by piece!

Thus, effective prioritization starts from defining a strategy. Answering the following questions helps to formulate it:

1. What do we care about most? Look first at events affecting the critical assets!

2. Has this attack succeeded? Look at the available context information and then decide which attacks have a higher chance of being successful and then chase first.
3. Has that ever happened before? Look at rare and new log message types first, as they ha a higher chance of being malicious.
4. Where is it coming from? DMZ and Internet-facing systems will likely be addressed first, as they usually are under highest threat
5. Is that truly an alert? Knowing what security devices produce false alarms helps to deprioritize them and save time for other activities.

The above will help to get you started on the prioritization strategy that will ease the burden of gigabytes of log data, possibly collected every day.

Even the most advanced and security conscious organizations fall into the pitfall of the fifth 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.** 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 thru 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 five mistakes we covered will take your log analysis program to a next level and enhance the value of the existing security and logging infrastructures.

Anton Chuvakin, Ph.D., GCIA, GCIH (http://www.chuvakin.org) is a Security Strategist with netForensics, a security information management company, where he is involved with designing the product, researching potential new security features and advancing the product security roadmap. His areas of infosec expertise include intrusion detection, UNIX security, forensics, honeypots, etc. He is the author of a book "Security Warrior" (O'Reilly, January 2004) and a contributor to "Know Your Enemy II" by the Honeynet Project (AWL,

June 2004) and "Information Security Management Handbook" (CRC, April 2004). In his spare time he maintains his security portal http://www.info-secure.org