OPTIMIZE LOG MANAGEMENT

How to collect and store logs securely
How to collect and store logs securely

The Situation
Log management is a cornerstone of good security practice as well as the underpinning of all compliance requirements. Your organization has asked you to ensure you comply with best practices for secure collection, transmission, and storage of logs.

The sheer wealth of potential information to collect across the enterprise means that you need to deal with billions of log events per day, across diverse sources from the network through to the operating systems and, ultimately, applications and databases. Not only will you need to architect a secure implementation, you must ensure that the solution scales and can meet the objectives of the business.

Driving Concerns
There are several fundamental factors to consider when it comes to secure log collection. For one, the most common native logging protocol, syslog, is a UDP protocol. UDP is “fire and forget,” without any acknowledgement from the receiving device that the data has arrived. As a result, syslog does not secure or guarantee transmission of log data.

In addition, logs must be treated with the respect given any other piece of evidence. Due care should be given to the collection and handling of this evidence, following the same standards a forensic officer would use for a piece of evidence in the physical world. Let’s consider what due care means for the world of logging.

• **Non-repudiation**—Non-repudiation is the science of ensuring and proving that the original source has not been tampered with. Logs need to be stored in such a way that non-repudiation can be proven. This proof is especially important when the need arises to present logs in the form of digital evidence.

• **Secure Chain of Custody**—To be accepted by a court as “good evidence,” logs must be handled according to exactly the same principles as physical evidence. The definition of Chain of Custody from Wikipedia is “the chronological documentation or paper trail, showing the seizure, custody, control, transfer, analysis, and disposition of evidence, physical or electronic.”

• **Privacy concerns**—Every region has privacy laws, and some are more restrictive than others. In a multinational corporation, it is important to understand each relevant country’s laws relating to how log data must be collected and handled. For example, some countries’ regulations forbid that any employee of a company see user-identifiable data without the express written consent of the company’s general counsel. This consent is usually only given when there is a strong suspicion that an employee is behaving in a malicious way.

• **Protocol limitations**—Syslog is arguably the oldest and most common protocol used in logging events. Unfortunately, it is also the least reliable, with default implementations relying on UDP transport mechanisms that do not guarantee delivery to log servers. Since you will probably use UDP, this limitation needs to be taken into account when planning a logging solution. The same is true of SNMP; its default configuration is to send traps via UDP, although this is less common as a logging source.

• **Beyond the operating system and network**—Many organizations log operating system and network activities and stop there. In addition, you need to consider collecting logs from applications (such as ERP systems and legacy applications) and databases. As well as being good security practice, compliance mandates ordinarily dictate collection of logs across everything in an enterprise. Custom applications, especially legacy applications, are more likely than not to have non-standard logs, or logs that are not sufficiently instrumented to meet audit requirements. The cost of instrumenting applications and developing custom parsers for applications whose use or misuse constitute high risk for the enterprise must also be factored into the logging project.
Optimize Log Management

- Give special attention to database log collection—Few things make a database administrator more uncomfortable than when the security team asks for auditing (logging) to be activated on their database servers. The reason for this is valid—switching on native auditing typically leads to a performance hit on the databases of up to 25%. This is not a statistic most companies are prepared to accept, so logging often remains turned off. This norm is contrary to both compliance requirements and security requirements, since databases often contain an organization’s most vital data.

- Ensure that log generation never stops and processes stay secure—if the logging stops, the evidence trail ends, so many attackers attempt to derail logging and other security systems. For example, an administrator can stop and start the event logging service with ease on Microsoft Windows platforms, and a root user can stop the syslog service on UNIX and Linux platforms. When planning a solution, you must carefully protect these processes from being either hijacked or stopped by malicious actors. You should also implement a notification process for when log collection fails due to any breakdown: hardware, software, or human involvement.

If your collection methodology can overcome these challenges, you will be in a stronger position to prove due care when it comes to the Chain of Custody and non-repudiation of collected log messages. You will want this confidence should the day arise when the logs need to be produced as evidence.

Solution Description

McAfee recommends splitting the log collection and storage process into four constituent parts: log generation; log collection and transport; log storage; and log analysis, forensics, and reporting. Let’s look at a breakdown and best practices for each step.

- Log generation—Operating systems, network devices, applications, and databases typically generate logs natively. For effective and secure log generation, we need to consider the following:
  » The processes that are used to write events on servers and desktops are fairly trivial to hijack, so they need to be protected. There are two considerations here: protecting the logging service itself and ensuring the integrity of the actual log files. To protect the logging service, McAfee recommends deployment of application control technologies to prevent users from stopping and starting, hijacking, or otherwise tampering with the logging service.
  » To ensure their integrity, the log files themselves should be protected by file integrity monitoring solutions with active prevention of changes. This means the actual log files can be restricted to allow only approved logging processes to write to the files—and no other user or application. This restriction will prevent the log files themselves from being tampered with by a malicious user (for example, someone deleting log entries to cover their tracks). These two protections together ensure the native logging service continues to function, even when under attack from a malicious source.
  » Special attention to database log generation. The solution should offer an alternative to native database auditing that does not add performance overhead on the box. Options include host-based or network-based database activity monitoring solutions. These solutions can generate logs based on database activity with minimal overhead.
  » Protocol considerations—Two strategies mitigate the lack of guaranteed delivery in UDP.
    - If your vendor supports it, switch on TCP syslog (sometimes referred to as syslog-ng), which is a more robust delivery methodology for log data. Syslog-ng is a more complex implementation then standard syslog as it offers more options, such as secure logging using the transport layer security (TLS) encryption protocol, log storage, relaying logs from multiple clients via a relay server, and secure storage.
    - If utilizing syslog-ng is not possible, place your log collectors as close to the log source as possible (at a minimum, on the same subnet) to greatly reduce the risk of lost syslog packets. As a rule, auditors are happy with these mitigation strategies, as they recognize the shortcomings built into the syslog standard. The same methodology also applies if collecting via SNMP over UDP. (See “Placement of log collection technologies” below.)
» **Real-time transfer of log data**—As much as possible, the solution should move the log data in real time to a centralized logging infrastructure. Moving the logs off the host promptly minimizes the risk of logs being overwritten, deleted, or corrupted. If real-time transfer of data is not practical, you should build a strategy that minimizes the amount of time logs remain on the host before being transferred over to a centralized logging infrastructure. Common tactics include setting up a batch job to SCP the files off the host, or creating collection scripts to pull the files from the source device as often as is feasible.

» **Considerations of timestamps**—Another often overlooked problem is one of inconsistent time stamps on servers, devices, and applications. When generated, logs receive the timestamp from the host. If the host’s time is incorrect, the logs will be incorrectly stamped and may be questioned as “good evidence” in court. All logging sources should point to a network time protocol (NTP) server for time synchronization. This is not always feasible, so the central logging infrastructure should also append its own timestamp to the log as metadata.

- **Log Collection/Transport**—Next, we consider how to collect the log data to ensure that we can prove secure Chain of Custody. The most common collection methods are:
  - Push technologies: Syslog, SNMP, eStreamer, SDEE, and OPSEC.
  - Pull technologies: WMI, SQL, SCP, FTP, and HTTPS.
  - Agent collection technologies.

The process around log data must be automated, consistent, and demonstrable. The solution should accommodate these factors:

» **Agentless versus agent-based**—Agentless is the easiest option for collecting log data using native protocols, but, as discussed above, this option can have challenges when using UDP-based technologies. However, there are secure and guaranteed agentless collection methodologies, such as SCP, HTTPS streaming, HTTPS download, syslog-ng, and others. Alternatively, agents allow for secure and guaranteed collection from the end device to the logging infrastructure. Bidirectional agent-based log collection also allows you to leverage the on-host agent for other activities not directly related to log management, such as remediation. In addition, agents are stateful and can be configured to collect logs from additional neighboring hosts, for example, tailing flat files to see the most recent part of a long entry or collecting application logs. When considering an agent, you must weigh the overhead of agent maintenance against the reduction in the risk of log data being lost.

» **Placement of log collection technologies**—When collecting from UDP sources, log collection devices should be placed as close as possible to the source to mitigate the risk of lost data. The ideal would be to reduce to zero the number of hops a packet needs to travel before reaching its destination. Log collection devices help here. In a distributed architecture, they can be placed in multiple locations. Once logs are received on the collection appliances, they can be “wrapped” in a TCP packet for guaranteed onward delivery to the rest of the logging infrastructure.

» **Handling of data once received**—Once received onto the logging infrastructure, the logs should be digitally signed immediately. This digital signature enables proof of non-repudiation. When transported to other component parts of the infrastructure, logs should be encrypted. If the logs were received from a device using the UDP protocol, a TCP wrapper should be put around the log files at the receiver to enable guaranteed delivery to other parts of the logging infrastructure.

» **Parsing**—Log data in its raw form is difficult to analyze given the unstructured nature and lack of common formatting. For this reason, logs should be parsed and stored with sufficient field indexing to facilitate rapid searches and relational data forensics.

» **Normalization**—Every logging device will present a unique set of event syntax and message details that will often vary greatly, even for a single vendor. Given this challenge, the enterprise logging solution must support an extensible taxonomy of event categorization such that events of a common nature can be referenced using indexed descriptions of a generalized nature. For instance, all events of type “login” should be mapped to a common normalized identifier, regardless of the original source of each authentication event. With this effort, an FTP login, a domain account login, an Exchange login, and an SAP login would all be assigned a similar normalized identifier at the time of collection. That identifier could later be used in forensic searches or event correlation rules.
» **Metadata**—Generally, metadata should not be added to the raw log files to avoid any appearance of tampering. However, it is generally accepted that appending a timestamp to the logs when received does not alter the data for the purposes of proving non-repudiation. Note that this will hold true only if the process can be proven and reversed. Adding this timestamp at receipt is necessary to mitigate the risk that the source device time may be set incorrectly.

» **High availability**—The log collection platform should support an optional architecture of high availability to ensure no loss of critical log data. This is of utmost importance when using a syslog-based collection process since no stateful connection exists to guarantee the delivery of event/log data from the source.

» **Auto-detection**—New log sources should be detectable via an automated process, allowing asset owners to configure log forwarding to a designated collector. Once new logs are detected by the collector, the logging administrator should be required to confirm the log source type before consuming new logs into the platform.

» **Distributed denial-of-service (DDoS) protection**—To avoid creating a DDoS against the logging system, unknown log sources should be prevented from streaming high volumes of data into the collection platform. Access control lists should be used and enforced to prevent unauthorized or abusive log broadcasting.

• **Log storage**—Logs need to be digitally signed to prove beyond all doubt that the logs could not have been tampered with once received.

  » **Compression**—Compression is acceptable when storing raw log data, as long as the process is demonstrable and reversible. Utilities like gzip are acceptable.

  » **Storage of digital signatures**—Store the digital signature away from the raw log files in an encrypted format. This practice reduces the risk of the log file being tampered with and the associated hash being replaced.

  » **Encryption of logs**—It is generally not necessary to encrypt the raw log data at rest if good user access controls are applied within the infrastructure to stop unauthorized users viewing the data.

  » **Retention**—Different compliance mandates require different periods of storage time for log data. Generally, an enterprise should plan for a year's worth of log data to be made available online (for immediate use and analysis) and an additional six years archived, with the ability to restore as needed.

  » **Data masking**—Often a source log may contain sensitive or proprietary information, such as credit card or Social Security numbers. The log collection and storage platform should support a configurable option to mask sensitive data in logs to obfuscate its viewing by the log analysis staff. Masking also protects private information that might be stored in logs on unencrypted file systems.

• **Log analysis, forensics, and reporting**—Once log data has been collected and stored securely, enterprises need to consider the best and fastest methodology for accessing that data. In even small environments, simple grepping through the log files is unacceptable due to the amount of time it would take to retrieve queries.

  » **Indexing**—Indexing is the key to executing fast queries against vast quantities of raw log data. As a rough rule of thumb when calculating storage, compression reduces the raw log file by at least 17 to one (depending on type). The index is roughly double the size of the compressed log file. When indexing, the additional storage overhead needs to be factored in.

  » **Correlation**—Event correlation is the method by which event patterns of known security risks or suspicious activities can be identified through the methodical analysis of logged event data. A solution with automated correlation identifies such patterns as a potential incident—a series of events that together represent a higher level of threat or risk to the enterprise. For instance, a system might alert on a possible brute force login attack based on this event sequence: “X number of login failures within Y number of minutes followed by a successful login.”

  » **Interface**—Log searches should be intuitive and include the ability to save multiple searches, key on results, and provide intelligence to assist forensic analysis.
» **Export of raw logs**—Ultimately, when presenting digital evidence, the solution needs to allow for the secure export of the raw log data (not the indexes, but the actual logs). Once exported, the digital signatures of the logs should be re-verified within the solution.

**Technologies Used in the McAfee Solution**

To fulfill these requirements, the McAfee® solution has two major components: McAfee Enterprise Security Manager and McAfee Enterprise Log Manager. For added protection, we also offer McAfee Integrity Control—application whitelisting and file integrity monitoring/change prevention.

![Diagram](image)

Figure 1. Distributed flat and hierarchical deployment models.

**McAfee Enterprise Security Manager and McAfee Enterprise Log Manager**

McAfee Enterprise Security Manager and McAfee Enterprise Log Manager work together to address the problem of secure log collection, analysis, and storage. McAfee Enterprise Security Manager offers powerful and scalable security information and event management (SIEM). This SIEM technology enables real-time visibility into the most critical incidents within your enterprise. McAfee Enterprise Security Manager’s sophisticated engine can perform real-time, high-speed correlation of events across your infrastructure, supplementing these events with content analysis (by looking into the data), context analysis (taking into account, for example, netflow data), and threat data from McAfee Global Threat Intelligence. This unique capability of event correlation supplemented by situational awareness enables you to have a true view of your threat profile in real time.

McAfee Enterprise Security Manager works in tandem with McAfee Enterprise Security Manager for an enterprise-scalable log management solution. McAfee Enterprise Log Manager will efficiently collect, compress, and store all log files. Its integration with McAfee Enterprise Security Manager provides advanced searching, analytics, correlation, alerting, and reporting. Because all events and alerts provide easy one-click access to the original source log record, your forensic efforts will benefit, too.

McAfee Enterprise Log Manager helps organizations ensure a secure Chain of Custody and non-repudiation of log data:

- **Log collection**—McAfee Enterprise Log Manager supports log collection into a collector appliance via all native logging protocols, including SNMP, syslog, Windows Management Instrumentation (WMI), and file transfer (ftp, scp, http/https), as well as more guaranteed collection protocols such as syslog-ng. In addition, the solution also includes host-based agents for secure, guaranteed collection from the Windows platform when either WMI or an event log to syslog translator (such as SNARE) is not deemed secure enough.

- **Digital signatures**—As soon as the log data is received on the collection appliance, it is digitally signed to prove it has not been tampered with.
• **Secure transfer**—After receipt on the collection appliance, the log data is transported securely through the infrastructure to its final resting place. UDP data (from syslog or SNMP) is wrapped in a secure TCP wrapper, and all data is encrypted for transport using AES.

• **Timestamp**—A timestamp is added as metadata to the logs when they are received to ensure time and date consistency even when the originating device's clock is wrong.

• **Optimized storage**—The raw log data is compressed at its final destination using industry standard gzip compression, enhanced to achieve 20:1 compression ratios. The digital signatures are stored separately from the raw log data within the infrastructure.

• **Multiple storage options**—The solution supports native storage onboard the appliances, in addition to CIFS, NFS, NAS, and SAN options.

• **Flexible retention options**—Log data can be stored online for as long as needed to meet internal company standards or external audit/compliance requirements. You can also archive log data offline.

• **Privacy support**—McAfee Enterprise Log Manager supports obfuscating certain configurable types of data. As an example, for those organizations operating in countries where user-identifiable information must not be viewed, McAfee Enterprise Log Manager can be configured to obfuscate user name and IP addresses from reports.

• **Log data support across the enterprise**—McAfee Enterprise Log Manager supports both known and unknown log types. You can supplement out-of-the-box support for all common operating systems, network equipment, applications, and databases with custom log support for internal applications.

• **Scalability**—McAfee Enterprise Log Manager is an appliance-based, fully distributed technology. As enterprises grow, additional appliances can be added into the architecture to handle additional volumes. You access all appliances via a common management interface. Further, the solution also supports failover capabilities to ensure redundancy within the logging infrastructure.

• **Speed**—By using a sophisticated indexing algorithm, McAfee Enterprise Log Manager searches are returned nearly instantaneously. This is tremendously important as the science of forensic analysis involves being able to interpret evidence and use that evidence to guide and inform new lines of enquiry.

• **Intuitive interface**—The McAfee Enterprise Log Manager interface makes it simple for the analyst to navigate, search, collate evidence, and run reports. To enhance situational awareness, the interface guides the analyst to other areas relevant to the investigation. For example, when doing forensic analysis on an incident, the interface allows the security operations center (SOC) analysts to rapidly pivot, slice, and dice the data to discover relationships between events.

**McAfee Integrity Control**

This solution integrates application control with file integrity monitoring and prevention technologies. Application control can prevent the event generation process or service being stopped, even by an administrative or root user. Further, it can log any attempt to do so. This is a key step, as a malicious user would first stop the logging service while executing his attack or theft, starting the service up again when finished. Without protection technologies in place, there would be a gap in the event data collected at the most critical time.

File integrity monitoring and change prevention technology preserve the integrity of the log files as they are being written to the host collection device. This technology will ensure that only the authorized operating system process can write to the log files, preventing the deletion or the injection of malicious entries. As with application control, this technology will also log any attempts to disrupt the process of log collection.
**Impact of the Solution**

Deployment of McAfee Enterprise Security Manager and McAfee Enterprise Log Manager addresses the need for secure log collection, transport, handling, and storage, permitting your organization to prove non-repudiation and Chain of Custody of log records. McAfee Integrity Control ensures that the native log files being generated are a true and accurate record, preventing malicious users from interfering with the log generation process. With the McAfee solution, you go a long way toward enabling both compliance and best practice requirements that emphasize the need for log management, and you can go a step further to complete real-time visibility into the most critical incidents within your enterprise.

Adding these capabilities to other technology areas—such as vulnerability management and risk management—you can get full situational awareness about your environment. As an illustration, log management and SIEM show which threats are occurring in real time in the environment, while vulnerability management will determine what risk those threats pose based on the security posture of your assets. Other elements of the McAfee Security Management portfolio, such as the McAfee Global Threat Intelligence for SIEM subscription, can be added to help you quickly understand the nature of changing threats. McAfee® ePolicy Orchestrator® (McAfee ePO™) software can knit together these components for easier, more consistent policy management and integration with countermeasures.

Although there is much to consider when implementing a secure log collection and storage project, McAfee has designed a solution that will automate the process, taking into account each step in the workflow. Compliant log management can become part of a seamless lifecycle that includes continuous monitoring, situational awareness, and countermeasure analysis. Instead of spending your time collecting data, you can invest your time interpreting and acting on the results.

**Additional Resources**

- www.mcafee.com/SIEM
- www.mcafee.com/ESM
- www.mcafee.com/ELM
- www.mcafee.com/IntegrityControl
- www.mcafee.com/ePO

For more information about the Security Connected Reference Architecture, visit: www.mcafee.com/securityconnected

**About the Author**

Darren Thomas is a senior product manager at McAfee, responsible for vulnerability management and discovery technologies. Thomas has more than 12 years experience in the IT security industry. He has held technical and commercial positions at a variety of security vendors, including LogLogic, NetIQ, PentaSafe, and IBM. Thomas has been at McAfee for four years, joining the product management team in 2012 after his engagement as enterprise solution architect lead for McAfee security management solutions in EMEA.