Comprehensive security logging in IT

An attempt to make sense of it

Table of Contents

Last changed: Mar 5, 2013

Introduction

This paper attempts to:

- Describe the current landscape of Information Technology when it comes to logging
- Summarize the requirements for logging in the context of security
- Propose a sensible logging infrastructure for creating, filtering, storage and processing

This paper concentrates on the security environment within IT, not the operational one. However, since both environments very often process logs from the same sources, the implementation proposal can also be used to provide a logging infrastructure for operational purposes.

Operational environment

Even though the operational environment and its needs for logging are distinctly different from the security environment and its needs, it still needs logs and a way of intelligently process them. Often, logs needed for operational purposes are created by the same devices or applications which also create logs for security purposes.

Information Technology must ensure that any IT asset – a server, a printer, a database, a web site, etc. - must always be available and perform at the best possible level. Failures must be detected and acted on as soon possible, trends must be established to predict future needs or bottlenecks, and managers want to see that all that money they spent was actually worth spending (often referred to as “return of investment”).

Security environment

Information Technology must often adhere to many different compliance standards. Some of them are dictated by international treaties or trade rules[1], some by national laws and regulations[2], some by state[3] or other more localized laws and regulations, some by industry wide standards[4], some by company internal rules and bylaws and some simply by best practices[5].

To make matters worse, IT must usually adhere to several, sometimes overlapping or even conflicting rules.

The only way to be able to adhere to these rules is to collect as many logs from as many “in-scope” sources, filter out the “noise”, store the meaningful logs and provide a mechanism to process them in a way that satisfy compliance standards, produce real-time incident reports as well as regular overview reports.

1 Example: ISO 20022 (used by SWIFT)
2 Examples: SOX (publicly-traded companies), GLBA (financial institutions), HIPAA/HiTech (health organizations), NIST (government agencies and utility companies)
3 Examples: Mass. 201 CMR 17.00, New York State Information Security Policy
4 PCI-DSS 2.x (anyone accepting or processing credit cards)
5 Example: ISO 27001/27002

Logs, logs and more logs

One of the biggest challenges for any meaningful logging infrastructure is the fact that an IT organization has either not enough or way to many logs. By default, a lot of devices and applications do not produce many logs – or have only very limited storage capabilities to retains logs for more than a day (sometimes even just a few hours). System administrators are often advised by IT security personnel (or guidelines) to enable more logging – which sometimes make things worse:

- Extended logging might impact performance (eg. on routers or firewalls)
- Extended logging requires even more storage (RAM and/or Disk)
- Extended logging often produces many unwanted information
- Extended almost always produces more information than a human being can handle

Even in a small IT environment with just one firewall, one email server, one file server, one main application server, a handful of printers and maybe 20 desktops, extended logging can easily overwhelm the devices that produce and store the logs – let alone the often "one-person" IT department.

Sources of logs

Logs are being produced by many different sources:

- Network devices (routers, switches, load balancers, access points, VPN concentrators ...)
- Security devices (network and application firewalls, IDS/IPS, video surveillance, …)
- Servers
- Printers
- Desktops, Laptops
- Applications (web site, database, ERP, CRM, …)
- ...

Identifying which of all these different sources produce logs that are meaningful in the context of IT security presents a challenge.

Common questions which can help determine whether a log source is "in scope", ie. whether it contributes meaningful log data are:

- Is the log source involved in the processing or protecting credit card holder data?[6]
- Is the log source involved in the processing or protecting of private health information?[7]
- Does the log source store or process confidential information?
- Does the log source transmit confidential data between systems?
- Does the log source control access to other IT assets?

6 PCI-DSS 2.x
7 HIPAA/HiTech

Using these questions as guidelines allows us to correctly identify the log sources that our infrastructure must collect data from.

Types of logs

Just as logs are created by many different sources, they can be presented in many different formats:

- Syslog as per RFC 5424[8]
- Windows Eventlog[9]
- Web server logs (Apache common log format[10], IIS log formats[11], …)
- Database logs (Oracle, MSSQL, MySQL, Postgres, DB2, …)
- SNMP traps
- …

Similarly, mechanisms for log transmission are plentiful:

- Syslog over UDP
- Syslog over TCP
- SNMP
- FTP
- Active MQ
- Jabber (instant message)
- …

Depending on the particular logging requirements, the logging infrastructure must support all formats and all transmission mechanisms that the log sources use.

This can often be accomplished by installing middleware or agents on or close to the log sources that can translate different formats into a common one and also accept different transmission mechanisms.

Contents of logs

While the overall format of logs is somewhat standardized (see above), the information contained in individual logs is VERY specific to the log source and can only be understood reading and understanding documentation provided by the manufacturer or through 3rd parties. In the case of open-source applications, the documentation is mostly available on the software main web site.

To create meaningful alerts and reports, the logging infrastructure should have a way of parsing and interpreting individual logs to extract meaningful information – which will often help in correlating logs from many, sometime disparate log sources.

8 https://tools.ietf.org/html/rfc5424
9 http://www.ultimatewindowssecurity.com/securitylog/encyclopedia/default.aspx?i=j
10 https://httpd.apache.org/docs/2.0/logs.html#common
11 https://www.microsoft.com/technet/prodtechnol/windowsserver2003/library/iis/bea506fd-38bc-4850-a4fb-e3a0379d321f.mspx

A big challenge lies in selecting logs what contents is meaningful for security monitoring and compliance. Logs should always be collected when they

- Describe (both successful[12] and unsuccessful) login events – from local networks and from the Internet or via VPN
- Describe logoff events – from local networks and from the Internet or via VPN
- Describe (at least write) access to protected or confidential data (incl. user credentials)
- Describe (at least write) access to objects or programs which access protected or confidential data (incl. user credentials)
- Describe modification of protected or confidential data (files, databases, etc.)
- Describe transmission of protected or confidential data (incl. user credentials)
- Describe addition of or changes to permissions and rights (incl. user credentials)
- Describe network or application attacks
- Describe anything unusual ...

Proposal for logging infrastructure

All well-defined logging infrastructure can “bring order to the chaos”. Its main components are:

- Central log storage
- Normalized and filtered logs
- Controlled web access
- Predefined real-time alerts and compliance reports
- Custom definitions for real-time alerts and compliance reports
- Local collection agents
- Protected logs

Central log storage

A central log storage system solves a number of challenges because it:

- Provides enough storage to store up to seven years[13] of logs
- Provides fault-tolerant storage[14]
- Provides storage of logs from many sources in one place for correlation
- Provides optimized access to search logs, eg. for forensic purposes

Typically, a central log storage system consists of

12 Passwords need to be obfuscated for successful logins
13 Best practice requires 12 months of online storage, and up to six years of offline storage
14 RAID 10 or mirrored SAN based storage

- Dedicated database server(s)
- Frontend server(s) which takes in logs from different sources (incl. local agents)
- Web server(s) for access (possibly coupled with an external Identity Management system)
- Reporting server(s) (for real-time alerts and regularly scheduled compliance reports)

Database server

While some or all of the servers listed above can theoretically reside on the same hardware, it is highly advisable to at least implement the database server separately (for security and performance reasons).

The database server should use a Hierarchical Storage Model (HSM) as its storage backend. High speed disks (or LUNs in case of a SAN) must be present to store the most recent logs (up to one week into the past), while slower speed disks (or LUNs) can be used for older logs. If desired, long-term storage (everything older than 12 months) can be stored on offline storage such as DVD or tape – these media can even be shipped to an offsite location provided they can be retrieved within 24 hours.

Depending on the number of log sources and the number of logs collected by this infrastructure, the individual servers might have to be designed with powerful and CPU, Memory and (disk and network) I/O throughput in mind. Similarly, redundant (disk and network) I/O paths might have to implemented.

In case of very high throughput requirements a load-balancing cluster of database servers should be used.

Access to the database server must be tightly controlled (see also below).

Normalized logs

The frontend server which takes in the logs and saves them into the database, must be able to understand all log formats that the "in-scope" log sources use. Local collection agents can often already translate log formats from different sources into a common format before transmitting the logs to the central server.

Similarly, the frontend server must be able to understand all log transmission mechanisms that the "in-scope" log sources use. Again, local collection agents can often already translate between different mechanisms and use a common mechanism to transmit the data to the central server.

The frontend server contains filtering capabilities. While it is also better to filter logs at the source – so that only meaningful logs are being transmitted – it might not always be possible to filter at all or to a degree that allows very fine-grained control. The frontend server can then filter out logs that do not need to be stored in the database since they constitute "noise".

Also, the frontend server must have write access to the database server. The connection between the frontend and the database servers must always be encrypted using high-grade ciphers[15] and protocols[16].

Any and all connections from log sources (or local agents) to the frontend server should also use encrypted connections – esp. when the data traverses a public or otherwise untrusted network. If the

15 AES-128 or better
16 TLS 1.x or better

log source can not natively provide encrypted communication, either a VPN or a local agent must be deployed between the log source and the frontend server.

Since local agents often have the capabilities to translate between different log formats and also understand several transmission mechanisms, it might be advisable to implement one (or several) of them directly in front of the frontend server so that the frontend server can concentrate on saving the logs into the database server.

In case of very high throughput requirements a local load balancer in front of a farm of frontend servers (or local agents) should be used.

Controlled web access

The web server allows security administrators, compliance officers and security analysts to access stored logs. Access to the logs must be tightly controlled and should be separated into a number of access roles:

- Administrators – have read-only access to ALL logs, manage (create, edit, delete, schedule) reports and alerts, create other accounts based on the defined roles
- Compliance personnel – have read-only access to all reports and alerts, manage (create, edit, delete, schedule) reports and alerts
- Security analysts – have read-only access to logs as defined by the administrators (eg. based on their security clearance)
- Operational personnel – have read-only access to logs as as defined by the administrators (eg. database logs for DBAs)

To protect the log data all accounts can only have read-only access to the logs themselves (see also below). The only account with "write" permissions to the logs is assigned to the frontend server.

The web server itself creates logs – these must also be stored in the central infrastructure.

In case of very high throughput requirements a local load balancer in front of a farm of web servers should be used.

Real-time alerts

The report server creates real-time alerts based on both predefined criteria as well as custom ones. By using regular expression syntax for the criteria, the report server can inspect each log individually as it becomes available in the database.

To facilitate real-time alerts, the report server must know:

- The criteria to use for an alert (predefined or custom)
- Where to send the alert to (eg. email address)
- How to send the alert (email, SMS, instant message)

A correlation engine to create alerts based on several logs from different sources or over a certain period of time could complement real-time alerts based on single logs.

Compliance reports

Besides real-time and/or correlation based alerts, the report server provides both predefined and custom compliance reports which

- Summarize daily compliance related events
- Are automatically sent via email[17] to interested parties (eg. compliance officer, CI(S)O, …)

To facilitate real-time alerts, the report server must know:

- The criteria to use for a report (predefined or custom)
- Where to send the report to (ie. email address)

Local collection agents

Local collections agents are either add-on software to servers (eg. SNARE agents) or dedicated appliances.

Dedicated appliances can be seen as a miniature version of the central frontend server without the database server connection – they contain the same capabilities to understand, handle and translate between different log formats and transmission mechanisms as well as filtering. They also will always use a common log format and an encrypted connection to communicate with the frontend server.

Additionally, collection appliances could implement filters similar to the one on the frontend server to reduce the amount of "noise" being sent.

Add-on software deployed on servers is capable to translate (and sometime also filter) log events from a mechanism used natively on the server to a common log format. Freely available syslog clients include:

Oracle audit logging[18] (native logging under Oracle)

Rsyslog Windows Agent[19] (supports encrypted syslog transmission)

Balabit Syslog-NG agent for Windows[20]

Eventlog to Syslog[21]

SNARE agents for Windows, Solaris, AIX, Lotus Notes, IIS, Apache, Squid, ISA, etc.[22]

Log protection

Logs are data – sometimes include confidential ones. So they must be protected the same way as any other confidential data in the network are protected. In general, they should be classified at least as "business/personal confidential" (see below).

Logs are routinely used in forensic investigations. A big part of any forensic activity is the assurance that the data are saved in their original form and have not been altered in any way since then.

17 Email should be sent via secure/encrypted channels
18 http://docs.oracle.com/cd/E19082-01/819-3321/audittask-11/index.html
19 http://www.rsyslog.com/rsyslog-windows-agent-2-0-released/
20 http://www.balabit.com/network-security/syslog-ng/central-syslog-server/features/windows-eventlog
21 https://code.google.com/p/eventlog-to-syslog/
22 http://www.intersectalliance.com/projects/index.html

Similarly, records must clearly show whether data has been moved (by whom, from where to where and at what time).

Log storage must therefore adhere to these requirements:

- Original logs must be stored “read-only” and must never be altered
- Original logs must be stored separate from any other data
- Any move of data (eg. from short to long-term storage) must be documented

Any non-log data like reports, configurations, credentials etc. can be on “read/write” storage, assuming that whatever other security requirements for these data are met.

Appendices

Data classification

Introduction

Whenever we talk about security in the context of IT, Software or more general, computers, we really talk about security of DATA.

Data represent living things (eg. Humans, animals, plants), things that occur in nature (eg. Mountains, rivers, oceans), artificial constructs (eg. Companies), ideas (eg. Patents, algorithms, manifests), …

Data is stored in many different ways (eg. Image, text, spreadsheet, database, book) and transmitted in as many ways (eg. Email, file transfer, letter).

The three main principles of data security can be summarized as Confidentiality, Integrity and Availability – in short CIA.

Confidentiality

Data contain confidential information, information that the source of the data wants or needs to keep private. When data are stored or transmitted, extreme care must be taken to not disclose all or any part of the data to unauthorized individuals or systems.

Examples:

- Do you want everyone to know about a stupid youth prank (that might even have lead to a conviction)?
- Do you want everyone to know that you won in the lottery?
- Do you want everyone to know your credit card number along with its PIN?

To ensure confidentiality, data must be classified into several classes, eg. “top secret”, “business/personal confidential”, “public knowledge”. Access to data must be restricted based on that classification – using unique IDs, passwords and other mechanisms. Data must be encrypted unless they represent public knowledge.

Incident response policies and procedures must be in place to handle any event when non-public data has been disclosed (accidentally or deliberately) – that must include legal (eg. “breach laws”) and contractual obligations.

Integrity

When data is stored or transmitted, it should not be altered in any way without detection. Similarly, it must always where the data came from and who handled it along the way (“chain of evidence”) and whether the data was collected correctly (“validity”).

Lastly, when data is being transmitted, it must be clear who sent it (“authenticity”) and the capability to hold each party to their obligations (“non-repudiation”) - that includes that neither party of a data exchange can deny having received or sent the data.

Examples:

- When transferring money from one account to another, the amount can not change
- Health information stored on computers can not change without attaching a history
- X-Rays or similar can not be changed when sent from the lab to your primary physician

To ensure integrity, data must be in immutable stores (when possible), eg. “read-only” volumes. Data must be accompanied by computed hashes which need to be checked for changes periodically (if the data is stored) or after transmission. Data must be accompanied by histories and/or logs which need to contain a description of the changes, who made the changes along with timestamps.

To ensure integrity in transmission, digital signatures or keys must be used to establish authenticity and non-repudiation.

Incident response policies and procedures must be in place to handle any event when data has been compromised – that must include legal (eg. “breach laws”) and contractual obligations.

Availability

Of course, data must be available when needed. When data become unavailable, mechanisms must be in place to detect such an event, alert whoever needs to know about it, log it and (if possible) automatically remedy the situation.

Examples:

- Email needs to be accessible at all times, even when the data store fails
- Stock-trading web sites must be accessible at all times – regardless of time zone
- Medical histories, allergies, known medical problems must be accessible at all times (think “car accident”)

To ensure availability, data storage and transmission paths should be setup in redundant fashion. The more “real-time” the need for the data is, the more this redundant setup must be “real-time” (eg. Instantaneous fail-over mechanisms).

Data must be available in case of natural (eg. Hurricanes, floods) or man-made catastrophes (eg. Power outage, hardware failure, system upgrade) or in case of attacks to the system (eg. “denial-of-service”).

Incident response policies and procedures must be in place to handle any event when data has become unavailable.

References

https://en.wikipedia.org/wiki/Information_security

http://it.med.miami.edu/x904.xml

http://www.doc.ic.ac.uk/~ajs300/security/CIA.htm

http://www.ncsl.org/issues-research/telecom/security-breach-notification-laws.aspx

Data definitions per standard[23]

Standard	Definitions
PCI[24]	Primary Account Number (PAN), or the 16-digit account number. PAN is the defining factor in the applicability of the PCI DSS requirements in 2.0, and is the element of cardholder data that must be masked, truncated, encrypted, or otherwise protected (per Requirements 3.3 and 3.4).
	Cardholder name
	Service code
	Expiration data
	Full magnetic stripe or equivalent chip data, which will contain elements of both the cardholder data and the SAD (Sensitive Authentication Data).
	PIN/PIN block, used in Chip-and-PIN cards primarily outside the US.
HIPAA[25]	An individual's past, present, or future physical or mental health or condition.
	The provision of health care to the individual.
	The past, present, or future payment for the provision of health care to the individual, and that identifies the individual or for which there is a reasonable basis to believe can be used to identify the individual. Protected health information includes many common identifiers (e.g., name, address, birth date, Social Security Number, personal vehicle information) when they can be associated with the health information listed above.
	Biometric data and full face images
	Payment Guarantor's information
GLBA[26]	Individual's or vendor Social Security Number (SSN) or Employer Identification Number (EIN)
	Financial account numbers
	Credit card numbers
	Date of birth
	Name, address, and phone numbers when collected with financial data
	Details of any financial transactions

23 http://www.sans.org/reading_room/analysts_program/encryption_Nov07.pdf, http://www.angelo.edu/services/technology/it_policies/dataClassificationStandard.php

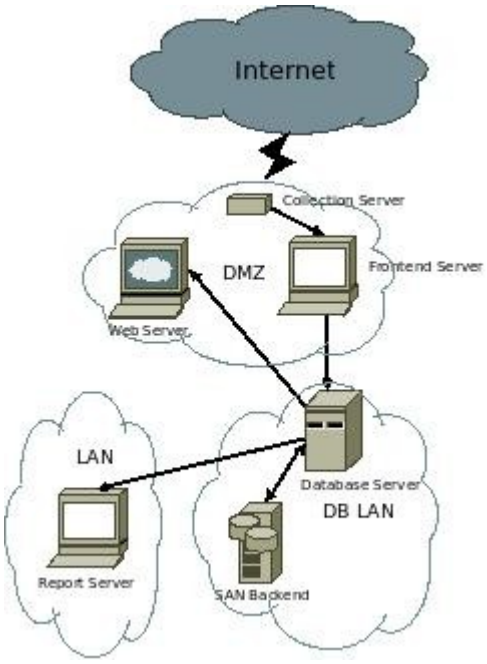
24 http://www.indefenseofdata.com/2011/03/pci-addressing-requirement-0-finding-cardholder-data/

25 http://www.hhs.gov/ocr/privacy/hipaa/understanding/coveredentities/De-identification/guidance.html#protected

26 http://www.itap.purdue.edu/security/policies/GLB_Safeguards_Rule_Training_Business_Office.pdf

Standard	Definitions
	Contract information
NIST 800-53[27] (and derived standards like FISMA[28] and FERPA)	Name, local and permanent mailing addresses, telephone number(s)
	Date and place of birth
	Marital status
	Photograph
	E-mail address(es) provided by the institution
	Previous (educational or other) agencies and institutions
	Access device numbers (building access code, etc.)
	FERPA: Residence assignment and room or apartment number, campus office address (for graduate students)
	FERPA: Student financials, credit cards, bank accounts, wire transfers, payment history, financial aid/grants, student bills
	FERPA: Dates of attendance; enrollment status; classification; degree(s); major and minor fields of study; awards and honors received
	FERPA: Hometown, parents' names and mailing addresses
	FERPA: Participation in recognized activities and sports; weight and height of members of athletic teams; team photographs

Example implementation



27 http://csrc.nist.gov/publications/nistpubs/800-53-Rev3/sp800-53-rev3-final.pdf
28 http://csrc.nist.gov/groups/SMA/fisma/overview.html

