IMPORTANCE OF EVENT LOG MANAGEMENT TO ENSURE INFORMATION SYSTEM SECURITY

Nicoleta STANCIU

Academy of Economic Studies Bucharest, Romania

===

Key words: event log, SIEM, log normalization

PhD Student, economist
Nicoleta STANCIU

Abstract: *This article presents an overview of the event logs as part of information system audit, their importance, how to analyze them to assess the effectivness of controls, the security log management what is the process for generating, transmitting, storing, analyzing, and disposing of computer security log data. It presents a generic Security Information and Event Management (SIEM) , tool that collect event logs generated by various sources, such as security software logs, operating system logs and application logs, correlate and analyze aggregated and normalized security event logs, in centralized manner. Also, it highlights idea to need the finding an interoperability standard format and new methods for log normalization.*

1. INTRODUCTION

The rapid expansion information technology in recent years has generated tremendous benefits to business and organizations, but at the same time the security-related threats, such as malicious code (e.g. virus, worm, Trojan horse), unauthorized system access, system intrusion, system penetration, distributed denial of service, have become not only more numerous and diverse but also more damaging and disruptive. The detection of security events is therefore necessary to minimize the loss and the destruction, to establish measures to mitigate the security risk.

Because of the widespread deployment of networked servers, workstations, and other computing devices has increased greatly and the number, volume, and variety of computer security logs, that has created the need for computer security log management.

Computer security log management [1] is the process for generating, transmitting, storing, analyzing, and disposing of computer security log data.

The **event logs** are part of the audit process and intrusion detection process. The audit process uses the **event logs** in the analysis to assess the effectiveness of controls.

The **event logs** are a series of records of system activities – operating system and application processes and user activities. So, there are system-level event logs, application- level event logs, user event logs. The terms [2] audit trail, audit log, activity log, event log and system log are synonymous. Event logs are obtained by monitoring information systems activity. The event logs are used for performing auditing and forensic analysis, supporting internal investigations, and identifying operational trends and long-term problems.

The **event logs** are used in information systems security [3] as follows:

Information system audit should be in accordance with C2 Auditing. C2 [21] is defined in the Trusted Computer System Evaluation Criteria (TCSEC).

A C2 system must be able to:

- provide system level audit trail;
- audit the use of identification and authentication mechanisms;
- audit file access (open, close, read, write, create) and program initiation;
- audit file/object deletion;
- audit administrative actions.

The widespread deployment of networked servers, workstations, and other computing devices and their diversity have led to increasing the number, volume and variety of computer security logs. This has created the need for events which occur on multiple systems to be correlated and analyzed together, thus can be detected events which would be impossible to detect if each system had a separate log.

2. TYPES OF EVENT LOG AND CHALLENGES

The main types of event logs are the following [1]:

- **Security software logs,** such as, antimalware software, intrusion detection and prevention systems, remote access software, web proxies, vulnerability management software, authentication servers, routers, firewalls;
- **Operating systems logs,** such as, system events logs, audit records;
- **Applications logs,** log various types o information from e-mail servers and clients, Web servers and browsers, file servers and file sharing clients, and database servers and clients.

Due to the fact that there are various event logs which can contain a wide variety of information on the events occurring with systems and networks , in log management should be consider the following issues:

- The event logs are generated by many sources and a source can generate multiple logs (e.g. UNIX), thus, an event can be recorded by one or more sources;
- various log formats – comma-separated or tab-delimited text files, databases, Extensible Markup Language (XML) and binary files;
- inconsistent log content - some logs contain detailed information but other logs contain less information about similar events; there are source which records only information that they consider most important;
- inconsistent mode of representing data – the sources record similar information differently, for example, a source records IP address and another name of computer, the date format is different, thus, MMDDYYYY and MM-DD-YY;
- setting timestamp – each host refers its internal clock, therefore requires synchronization.
- need to ensure the confidentiality, integrity, and availability of event logs at the source, during collection and at centralized storage system.

All challenges occur because each system has its own event log format.

3. SECURITY INFORMATION AND EVENT MANAGEMENT

Security Information and Event Management (SIEM) is a tool that collect event logs generated by various sources, such as security software logs, operating system logs and application logs, correlate and analyze aggregated and normalized security event logs, in centralized manner. The centralized mode allows by correlating the events by various sources that could by itself signify nothing(e.g. a syslog message from a firewall), but in combination with events from another source (e.g. a syslog message from UNIX server, a message from a router), can identify attacks that can cause great damage.

For an efficient and effective log management the organizations must perform the following issues([1], [4]):

- Establish policies and procedures for log management;
  - Document which assets are in scope for each compliance regulation of the organization;
  - Define which networks and assets are taken into account;
  - Define the Events of Interest (EOI), that could constitute a threat (anexa A [4]); Of the millions of events per day an organization collects, less than 1% will represent a threat;
  - Define and document Service Level Agreements (SLAs);
  - Define and document Standard Operating Procedures (SOPs) for every tiers of the management process - configuring log sources, performing log analysis, initiating responses to identified events, and managing long-term storage.
- Prioritize log management appropriately throughout the organization - based on the organization’s perceived reduction of risk and the expected time and resources needed to perform log management functions;
- Create and maintain a log management infrastructure - hardware, software, networks, and media used to generate, transmit, store, analyze, and dispose of log data;
- Define roles and responsibilities for log management for key personnel throughout the organization, including establishing log management duties at both the individual system level and the log management infrastructure level;

A log management infrastructure typically comprises the following tiers:

- Log Generation
- Log Collection
- Log Storage
- Log Analysis and Incident Response
- Monitoring

Every tier can vary greatly in complexity and structure, being specific to each product.

Main followed aspects refers to: status of finished activities versus schedule; quality of work done, project team attitude and 3rd party involved into project (including clients and management) [22].

3.1. Log Generation

In the log generation process should be considered the following aspects [1]:

- types of assets which should perform logging;
- the components which should perform logging (e.g., OS, application);
- types of events which should log;
- frequency of logging.

Logs can contain information on the events occurring within systems and networks. Every system has its own event log format. Most of the sources have a monitoring system that run continuously, so they generate entries in log files on an ongoing basis. In order to achieve a centralized analysis it requires a good understanding of all formats.

I will present below some ways of logs generation.

3.1.1. Windows Event Logging Framework

EVTX is Microsoft Windows log format implemented in Windows Vista, Windows Server 2008 and Windows 7.

The event properties defined by Microsoft are presented in **Table 1**. Event Properties [5]:

The following properties can be adding to the Event Viewer: Process ID, Thread ID, Processor ID, Session ID, Kernal Time, User Time, Processor Time, Correlation ID, Relative Correlation ID.

Windows Vista includes two categories of event logs [6]: **Windows Logs** and **Applications and Services Logs**.

The **Windows Logs** category includes: Application, Security, System logs, Setup log and the ForwardedEvents log.

The **Application** log contains events logged by applications or programs. For example, for *MSSQL$SQLEXPRESS source, Recovery is complete.*

The **System** log contains events logged by Windows system components. For example, *An error was detected on device \Device\CdRom0 during a paging operation.*

The **Setup** log contains events related to application setup.

Table 1. Event Properties

Property Name	**Description**
Source	The software that logged the event
Event ID	A number identifying the particular event type.
Level	A classification of the event severity. The following event severity levels can occur in the system and application logs: **information, warning, error, critical, success audit, failure audit.**
User	The name of the user on whose behalf the event occurred.
Operational Code	Contains a numeric value that identifies the activity or a point within an activity that the application was performing when it raised the event.
Log	The name of the log where the event was recorded.
Task Category	Used to represent a subcomponent or activity of the event publisher.
Keywords	A set of categories or tags that can be used to filter or search for events. Examples include "Network", "Security", or "Resource not found."
Computer	The name of the computer on which the event occurred.
Date and Time	The date and time that the event was logged.

The **Security** log contains events which are audited, configured in *Computer Configuration\Windows Settings\Security Settings\Local Policies\Audit Policy*

Account logon events - logs an event when a user attempts to log on

Account management – logs an event when an account is managed

Logon events - logs an event for logon events that are occurring over the network or generated by service startup

Object access - logs an event when a user attempts to access a printer or shared folder and file, for each of the resource that you want audited

Policy changes - logs an event when a policy is successfully or unsuccessfully changed

Privilege use - logs an event when a user attempts, successfully or unsuccessfully, to use special privileges

Process tracking - logs an event for each program or process that a user launches while accessing a system

System events - logs designated system events, such as when a user restarts or shuts down a computer

The **Applications and Services logs** [6] are a new category of event logs. These logs store events from a single application or component rather than events that might have system wide impact. This category of logs includes four subtypes: Admin, Operational, Analytic, and Debug logs. Events in Admin logs are using the Event Viewer to troubleshoot problems. Events in the **Operational log** are useful to require more interpretation. **Analytic logs** store events that trace an issue and, often, a high volume of events are logged. **Debug logs** are used by developers when debugging applications.

For every type of event logs there is a **channel** that is [7] a named stream of events that transports events from an event publisher to an event log file, where an event consumer can get an event.

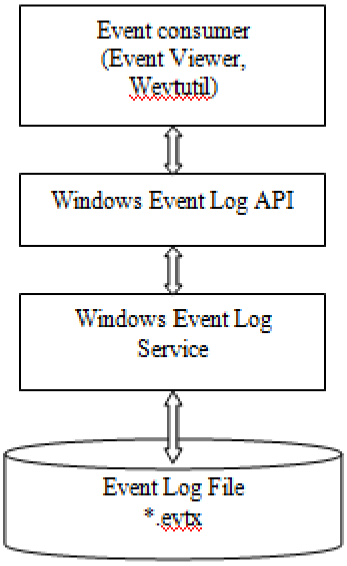


Fig. 1. Windows Event Log

The Windows Event Log service [8] exposes functions that enable programs to maintain and manage the event logs, configure event publishing, and perform operations on the logs, such as archiving and clearing.

The API [9] includes the functions that an event consumer, such as the Event Viewer, would use to read and render the events. To write the events defined in the manifest, use the functions included in the Event Tracing (ETW) API. An instrumentation manifest identifies your event provider and the events that it logs.

EVTX logs are stored in binary XML format files with .evtx extension. They can be exported in XML, TXT and CSV format. Still, we have an XML representation of an event:

```
<Event
xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
    <System>
              <Provider Name="Microsoft-Windows-Security-
              Auditing" Guid="{54849625-5478-4994-a5ba-
              3e3b0328c30d}" />
             <EventID>4776</EventID>
             <Version>0</Version>
             <Level>0</Level>
             <Task>14336</Task>
               <Opcode>0</Opcode>
              <Keywords>0x8020000000000000</Keywords>
               <TimeCreated SystemTime="2012-08-
              02T08:11:35.038Z" />
              <EventRecordID>2613806</EventRecordID>
               <Correlation />
              <Execution ProcessID="792" ThreadID="1756" />
             <Channel>Security</Channel>
               <Computer>USER-GQ11CYGM3S</Computer>
             <Security />
     </System>
       <EventData>
               <Data Name
              ="PackageName">MICROSOFT_AUTHENTICATI
              ON_PACKAGE_V1_0</Data>
               <Data Name
              ="TargetUserName">Administrator</Data>
             <Data Name="Workstation">USER-
              GQ11CYGM3S</Data>
             <Data Name="Status">0x0</Data>
        </EventData>
</Event>
```

The Event ID relationship for most security related events is EVTXEventId = EVTEventId + 4096

EVT log files are specific to Windows XP, Windows 2003 Server.

Some of the important windows events are listed below [10]:

- Local logon attempt failures: 4625, 4626, 4627, 4628, 4629, 4630, 4633
- A user account was created - 4720
- A user account was deleted – 4726
- Audit policy change: 4719, 4721, 4708
- Windows is starting - 4608
- Windows is shutting down – 4609

3.1.2. The UNIX System Log Facility – syslog

Many versions of UNIX provide a general-purpose logging facility called ***syslog*,** originally developed at the University of California at Berkeley for the Berkeley *sendmail* program and improved by the syslog protocol RFC 5424.

A centralized logging process runs the program */etc/syslogd* or */etc/syslog* that is host-configurable.

The syslog message has the following definition [11]:

SYSLOG-MSG = HEADER SP STRUCTURED-DATA [SP MSG]

HEADER = PRI VERSION SP TIMESTAMP SP HOSTNAME SP APP-NAME SP PROCID SP MSGID

PRI – Priority = Facility x 8+Severity.

Facility values are in the range of 0 to 23 inclusive, 0- *kernel*,1- *user*, 2-*mail*, 3-*system daemons*, 4/10- *security/authorization messages*, 5- *messages generated internally by syslogd*, 6- *line printer subsystem*, 7- *network news subsystem*, 8- *UUCP subsystem*, 9/15- *clock daemon*, 11- *FTP daemon*, 12- *NTP subsystem*, 13- *log audit*, 14-*log alert*, 16-*local0…*23-*local7* – reseved for site-specific use.

Severity values are in the range of 0 to 7 inclusive, 0- *emergency*, 1- *alert*, 2-*critical*, 3-*error*, 4-*warning*, 5-*notice*, 6- *informational*, 7-*debug*.

HOSTNAME – identifies the machine that generated the syslog message and contains the hostname and the domain name represented in FQDN, hostname, static/dynamic IP address.

APP-NAME - identifies the device or application that generated the message

PROCID – process name or process ID associated with a syslog system

MSGID - identifies the type of message. For example, a firewall use "TCPIN" for incoming TCP traffic

STRUCTURED-DATA- provides a mechanism to express information to support data analysis. This can contain zero, one, or multiple structured data elements – SD-ELEMENT. An SD-ELEMENT consists of a name – SD-ID and parameter SD-PARAM, name-value pairs.

MSG - contains a free-form message that provides information about the event

For example :

```
Apr  4 13:32:43 av postfix/smtpd[8328]: lost connection after
CONNECT from unknown[221.180.15.219]

Apr  4 06:25:12 av rsyslogd: [origin software="rsyslogd"
swVersion="4.6.4" x-pid="872" x-info="http://www.rsyslog.com"]
rsyslogd was HUPed, type 'lightweight'.
```

In these examples, we have: creation date - Apr 4 06:25:12, host name – av, app-name, and message.

Syslog can be used to integrate log data from many different types of systems into a central repository for information system management and security auditing.

Windows Event Log and other log formats can be converted to syslog.

3.2. Log Collection

Log Collection is the process of getting the data from the source device into the SIEM.

In this process in the logging policies should be established the following aspects([1], [12]):

- the devices from which will collect the events;
- the events which will collect;
- where the logs will store;
- how the logs will be transferred;
- frequency transfer - real-time, every 5 minutes, every hour;
- how to ensure the confidentiality, integrity, and availability during transfer.

There are two methods of log collection [12]: push method and pull method.

In the **push method** the source device sends logs to the SIEM an ongoing basis without any interaction from the SIEM itself.

An implementation of **push method** is by **syslog** protocol, in which each source device uses syslog protocol for its log and for transferring its log entries to a syslog server running on another host. The source device is configured by set up the IP address or DNS name of a syslog server on your network, where to be sent its logs. This method has the advantange that the many sources use syslog to log generation(Unix, Linux) and it is easy to setup and configure and the disadvantages when use UDP syslog to transfer logs between hosts, because UDP transport protocol is connectionless and the most syslog implementations use not encryption to protect the integrity or confidentiality of logs in transit. To remove this disadvantages it is recommended the use of the Transport Layer Security (TLS) protocol to protect the confidentiality of transmitted syslog messages [11] and passing syslog messages through secure shell (SSH) tunnels [1] .

In another implementation, the log files is written on the source device and this push their logs to the server, which usually involves each source device will establish a connection via secure protocol to the server, authenticating to the server and transferring its logs regularly.

In the **pull method** the log file is written on the source device, placed into a directory that has read permissions for SIEM. The SIEM will establish a connection on a regular interval, via a secure protocol to the directory using credentials, will collect the log file from source device and will write locally. The disadvantage is that this method may be batched to run at certain time periods.

From another perspective of log collection can be [1]: agentless or agent-based.

In **agentless** case it is not necessary to have any special software installed on the source device. Log filtering, log normatization and log conversion are performed at the centralized level.

In **agent-based** case an agent program is installed on the source device to perform log filtering, log normalization and log conversion for a particular type of log, then transmit the normalized log data to an SIEM server, usually on a real-time or near-real-time basis. The filtering and reduction proccess leading to decrease the amount of data retrieved through the network.

Because, in a complex system there are various types of logs, the log collection could perform by multiple methods, the SIEM could have different types of connectors to collect the logs from the devices. For example for Cisco ASA and Linux server it can use syslog, for Windows Server 2003, Windows Server 2008 there are more complicated solutions to convert Windows event log to syslog and is easier to pull the Windows event log from Windows Server to SIEM.

3.3. Log Storage

The are more types of log storage [13]:

- many systems store their logs by transfer over the trusted network to a central logging server in syslog format/specific interoperability format/database;
- remote storage – when there are multiple branch offices interconnected by Internet network, it performs a central logging at the each branch office level and relay to a second-level central logging by transfer over a non-trusted network (Internet).

The logging requirements of log storage level are [1]:

- establishing the necessary storage space;
- establishing how long each type of log should be preserved;
- establishing the time then unneeded log should be disposed of;
- frequecy of log rotation;
- preservation of the confidentiality, integrity and availability while storage.

To make analysis easier by using a single type of analysis engine and to reveal incidents that can only be performed by events correlation from multiple sources it is necessary to store the logs in a **standard format** with consistent fields representation which contains all neccesary information.

There are several storage strategies of event logs at centralized logging level:

- all raw event logs only in its own format;
- the raw event logs converted into standard format specific to every SIEM;
- the raw event log store in a field of standard format;
- the both, the raw event log and the event log in standard format, separately;

3.3.1. Attempts to Standardize a Event Log Interoperability Format

Intrusion Detection Message Exchange Format (IDMEF)

This standard is described in RFC4765 and it is experimental. The purpose of it is [14] to define data formats and exchange procedures for sharing information of interest to intrusion detection and response systems and to the management systems that may need to interact with them.

The data model is implemented in XML. The IDMEF data model is an object-oriented representation of the alert data.

Note that this model is complex and specific to the Intrusion Detection and Prevention Systems. It includes information about the event occurred, preparation of the information, analysis and evaluation in order to issue the alert.

Common Event Format(CEF)

Common Event Format is developed by ArcSight, used in ArcSight products, such as Enterprise Security Manager (ESM) and supported by several other products. This defines a simple event format that contains the most relevant information. It use syslog as a transport mechanism, that adds as prefix to the message date and hostname.

Sep 19 08:26:10 host *message*

The message is composed of fields delimited by a bar (“|”) character [15].

CEF: *Version|Device Vendor|Device Product|Device Version|Signature ID|Name|Severity|Extension*

```
Sep 19 08:26:10 host
CEF:0|security|threatmanager|1.0|100|detected a \\ in
packet|10|src=10.0.0.1 act=blocked a \\ dst=1.1.1.1
```

Extension is a collection of key-value pairs [15].

Common Event Expression (CEE)

The Common Event Expression (CEE) is an open event log standard developed by MITRE [16]. The CEE includes three components: CEE Profile, CEE Log Syntax and CEE Log Transport.

In [17] I proposed an own approach of event log format. In the model, an event is defined by three elements: **Subject, Action, Object**, it means that one or more subjects perform one or more actions on one or more objects.

- The subject that performs the action can be user, computer (server or workstation) or network device. The user can be system user or application user. The user’s privileges can “Administrator”, “User” or “Guest”. For example, Unix “root user” would be identified with “Administrator”.
- The object on which the action has been performed can be a file, a database element, a computer, a network device.

- The action can include: create, read, update, delete, execute, exit, authorize, initiate or accept network connection. .

  The data model can be object-oriented model or relational model.

3.3.2. Conversion to event log interoperability format

Log normalization is the process of transforming the different original event log formats to one of event log interoperability format describe above.

For log normalization you would have to follow these steps (inspired by [18]):

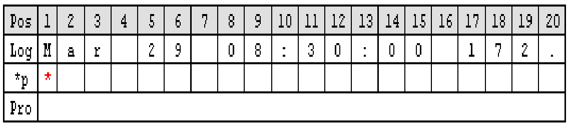
- *Read original event log record*; It should that each type of source device to store own event logs in central logging server in separate directories, because each event log record have its own format;
- *Extracts fields names and field values from original event log record* using specific rules to each format – called parsing log messages; The types of event log format have different structures, so, there are necessary more types of log parser. In order to parse a event log record it is vital to know its exact structure, the items of structure and the position of each item in the structure. Many devices (Unix servers, Linux servers, routers, firewalls) use syslog for logging but the content of their respective log messages are very different.

  Post-Process action of **Adiscon's MonitorWare** [19] provides an editor for creating **a log format template** for syslog message that consists of as many rules as necessary to parse out the relevant information.

  It takes for example the following message:

  Mar 29 08:30:00 172.16.0.1 %Access-User: 12345: rule=monitor-user-login user=Bob status=denied msg=User does not exist

The message is splitting used the following procedure:



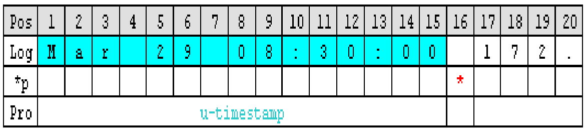
Pos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Log	M	a	r		2	9		0	8	:	3	0	:	0	0		1	7	2	.
*p	*																			
Pro																				

Fig. 2. Parsing log message, step one(source [19])

Pos = Position of the character.
Log = Message subdivided into its characters.
*p = Points to the position the parser stands after parsing the rule.
Pro = Property. In the term of Adiscon a property is the name of the item which is parsed out.

At the beginning of the parse process the parser's pointer points to the first character. Each parse type starts parsing at the current position of the pointer.

The first item is a Unix/Timestamp (it has a length of 15 characters) and it uses 'UNIX/LINUX-like Timestamp rule. It extracts u-timestamp field and moves the pointer to the next character after the timestamp.

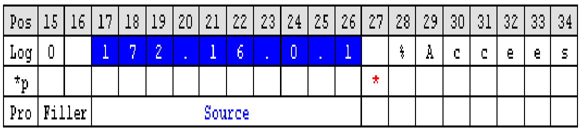


Pos	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Log	M	a	r		2	9		0	8	:	3	0	:	0	0		1	7	2	.
*p																*				
Pro	u-timestamp																			

Fig. 3. Parsing log message, step two (source [19])

After the timestamp is a space and uses 'Character Match' rule with a space as value.

The second item is IP-Address and it uses 'IP V4 Address' rule and it extracts Source fields.



Pos	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
Log	0		1	7	2	.	1	6	.	0	.	1		%	A	c	c	e	e	s
*p													*							
Pro	Filler		Source																	

Fig. 4. Parsing log message, step three (source [19])

The procedure continues until all fields are extracted.

- *Normalization field values* – the field values from original event log record are mapping into a standardized schema. The most common uses of normalization is storing dates and times in various formats (2:34:56 P.M. , 14:34) in a single format used time synchronization technologies such as Network Time Protocol (NTP). Another example is translate syslog severity.
- *Identify the exact meaning of the event*, for some formats (e.g., logon or logoff event);

  For example, Event ID 680 in Windows XP, Event ID 4776 in Windows Vista, ASA-sys-6-605005 in Cisco ASA means user login.
- Ensure that all fields required by the target format are present and drop fields that are present in the source but cannot be represented in the target;
- *Mapping the fields of original event log to the fields of event log interoperability format*;
- *Write the record in event log interoperability format.*

3.4. Log Analysis and Incident Response

Log Analysis is the process for studying log entries to identify events of interest in central logging. At this level, there are event logs of all source devices, stored in event log interoperability format, taking advantage to need a single analysis engine for all types of source devices. Generally, log analysis use rules, scripts or patterns to identify security incident, policy violations, fraudulent activity, and operational problems.

In this procces at least the following requirements to perform the policies and procedures would be set [1]:

- frequency analysis of the event logs;
- access permissions to the event logs;
- mode of action when suspicious activity or an anomaly is identified;
- how to ensure the confidentiality, integrity, and availability of the results of log analysis.

One of the operations at this level is **categorization.** The event logs can be order in categories accordind to the following elements:

- event identifier;
- source or destination IP address (e.g., can be compared with source address on a blacklist, destination address of a critical system);
- severity grade of the events (e.g., fatal, critical, error, warning, info);
- when creating - time of day or day of the week (e.g., an event might be acceptable only during certain times), ;

According to these categories it can assign **priorities** [1] to each type of event log (e.g., high/medium/low, 1 to 10).

Duplicates removal is another operation that removes the duplicate event log entries if same event is recorded by two or more source devices.

An important operation is **aggregation** – that creates a new, more generic event entry from several dissimilar event. Also [10], similar entries are consolidated into a single entry containing a count of the number of occurrences of the event.

Event-linking – this operation performs if sequence of events suggest particular scenario.

All these operations are used in **event correlation** [1] – is the operation which matches multiple log entries from one or more source devices based on logged values, such as timestamps, IP addresses, and event types. Event correlation is performed to find complex events, which can be highlighted only by combination more events from one or more source devices.

For example, below presents a possible brute force attack.

The analysis engines use a rule, to detect the security event.

If [count(type=failure)>=3 and type=success from same source and same destination and (T_f-T_i)<5 s] then attack

To detect attacks can be use many methodologies: rule-based, model-based, pattern matching, state transition analysis.

Table 2. Event Log

Type	Date	Time	Source	Destination	Category	Event
Success Audit	12.07.2012	11:58:03	172.45.25.20	172.45.25.10	Account Logon	680
Failure Audit	12.07.2012	11:58:02	172.45.25.20	172.45.25.10	Account Logon	680
Failure Audit	12.07.2012	11:58:01	172.45.25.20	172.45.25.10	Account Logon	680
Failure Audit	12.07.2012	11:58:00	172.45.25.20	172.45.25.10	Account Logon	680

4. LOG MANAGEMENT IN CLOUD COMPUTING

The Log Management in Cloud Computing [20] presents some facilities compared with the in-house Log Management, but also issues that complicate work.

Thus, there is significant increase of encryption during transport in log collection the rest remain unchanged and also, log normalization is unchanged.

At the log storage level it is no need to ensure storage capacity, log rotation to keep the size of log files manageable and log archival on removable media or SAN, they remain in the responsibility of cloud computing provider. In the in-house solution ensure the confidentiality, integrity and availability of event log is done locally and when the event log are moving to a cloud computing, the event log security is done by cloud administrator, and the client of cloud computing can not verify. There is a possibility that an attacker can extract and expose the data and also, that the event log accidentally intermingle with that of another cloud customer. It is good the the administrator of cloud computing can not view the data.

To perform the event log security in cloud computing the data is encrypted at the application or system level, and keyed at the organization, then the event log will not be viewable by anyone outside the organization even if they gain access to the cloud computing.

5. CONCLUSION

Log management seems to be a simple solutions but has become quite complicated due to varied sources of information, each sources with its own event log format. In recent years organizations realized the importance of the log management because they can find complex events by combining events from one or more source devices. It is seeing that the organizations begin to use the logs for more advanced purposes. The most organizations that use log management are financial and governement. They use to tracking suspicious behavior, in compliance process control, forensics analysis, security of network devices. The most organization are collecting logs from operating systems, followed by switch, router and firewall.

The mechanism of collection, storage and maintenance the event logs are no longer a problem due to the existence of large capacity of storage. It is still difficult the operations of log normalization, log analysis and reporting. Data storage into consistent standard format facilitates the analysis, such analysis engine can handle events from different systems using the same technique. Therefore, finding an interoperability standard format is a challenge, there is still motivation for research. Also, the operation of log normalization for transforming the different original event log formats to one of event log interoperability standard format must be improved and adapted to new standard format.

6. REFERENCES

[1] K. Kent and M. Souppaya, “Guide to Computer Security Log Management”, *NIST,* 2006

[2] K. Price, “Host-Based Misuse Detection and Conventional Operating Systems Audit Data Collection”, 1997

[3] NIST, “Audit Trail”, *National Institute of Standards and Technology,* 1997

[4] D. Swift, “Successful SIEM and Log Management. Strategies for Audit and Compliance”, *SANS Institute*, 2010

[5] Microsoft TechNet, “Event Properties”, 2011. Available: http://technet.microsoft.com/en-us/library/cc765981.aspx

[6] Microsoft TechNet, “Event Logs”, 2012. Available: http://technet.microsoft.com/en-us/library/cc722404

[7] Microsoft TechNet, “Event Logs and Channels in Windows Event Log”, 2009. Available: http://msdn.microsoft.com/en-us/library/aa385225.aspx

[8] Microsoft TechNet, “Windows Event Logs Services”, 2011. Available: http://msdn.microsoft.com/en-us/library/

[9] Microsoft TechNet, “Windows Event Log”, 2012. Available: http://msdn.microsoft.com/en-us/library/

[10] Microsoft TechNet, “Chapter 4: The Member Server Baseline Policy”, 2003. Available: http://msdn.microsoft.com/en-us/library/

[11] R. Gerhards , “The Syslog Protocol, RFC5424”, *Network Working Group,* 2009

[12] D. Miller, et. al., “Security Information and Event Management (SIEM) Implementation”, *The McGraw-Hill Companies*, 2011

[13] P. Matulis, “Centralised Logging with rsyslog”, *Canonical,* 2009

[14] H. Debar , D. Curry , B. Feinstein , “The Intrusion Detection Message Exchange Format (IDMEF)”, *IETF Trust,* 2007

[15] ArcSight, “Common Event Format: Event Interoperability Standard. ArcSight Technical Note”, *ArcSight, Inc.,*2009

[16] CEE, “A Standardized Common Event Expression (CEE) for Event Interoperability”, *The MITRE Corporation,* 2012, Available: http://cee.mitre.org

[17] N. Stanciu, *Event Log Interoperability Formats. Need to Develop,* The Eleventh International Conference on Informatics in Economy, 2012

[18] R. Gerhards, “Log Normalization Systems and CEE Profiles”, *Adiscon GmbH,* 2011

[19] M. Mekelein, “Parsing Log Messages”, *MonitorWare, 2006*

[20] T. Chmielarski, “Moving Log Management to a Cloud Computing Provider”, *GlassHouse Technologies*, 2011

[21] Departament of Defense, *Trusted Computer System Evaluation Criteria*, 1985

[22] Grigorescu, Carmen Judith, “Investment decision modelling in the microeconomic context” , Pro Universitaria Publishing House, Bucharest, 2011

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.