



Certified Network Defender v3
MODULE 15
NETWORK LOGS MONITORING
AND ANALYSIS

EC-Council Official Curricula

This page is intentionally left blank.

LEARNING OBJECTIVES

The learning objectives of this module are:

- LO#01: Understand logging concepts
- LO#02: Discuss log monitoring and analysis on Windows systems
- LO#03: Discuss log monitoring and analysis on Linux systems
- LO#04: Discuss log monitoring and analysis on Mac systems
- LO#05: Discuss log monitoring and analysis on firewalls
- LO#06: Discuss log monitoring and analysis on routers
- LO#07: Discuss log monitoring and analysis on web servers
- LO#08: Discuss centralized log monitoring and analysis

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Learning Objectives

To enhance the security of an organization, extensive monitoring and analysis of network logs is critical. This helps identify and respond to threats quickly and protect the network assets from various attacks. Proper network log monitoring and analysis help reduce the frequency of attacks by proactively responding to threats. The objectives of this module are listed below:

- Logging concepts
- Log monitoring and analysis on Windows systems
- Log monitoring and analysis on Linux systems
- Log monitoring and analysis on Mac systems
- Log monitoring and analysis on firewalls
- Log monitoring and analysis on routers
- Log monitoring and analysis on web servers
- Centralized log monitoring and analysis



LO#01: Understand logging concepts

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

LO#01: Logging Concepts

The objective of this section is to explain the basic concepts of logging. It describes the sources of logs, need of logging, log formats, and various logging approaches.

Log

- Log is a **collection of information/data on events** generated in the form of audit trail by the various components of information system such as network, applications, OS, service, etc.
- Logging is the process of **recording** and **storing logs** of the events that occur in the network
- It is an important source that helps detect flaws or problems as well as network attacks, frauds, and inappropriate uses of data

Example of Log:

- Trail of Login Failure events followed by Login Successful event

Typical Log Sources

The diagram illustrates various log sources across different network segments. On the left, the 'Executive' segment includes Client & File Server Logs, Windows Domain Logs, Windows Logs, and Wireless Access Logs. The 'Data Center' segment includes NAS Access Logs, VLAN Access & Control Logs, and Mainframe Logs. The 'Financial' segment includes Database Logs. The 'Research & Development' segment includes Linux, Unix, Windows OS Logs. The 'DMZ' segment includes Content Management Logs. The 'Internet' segment includes VPN Logs, IPS/IDS Logs, Firewall Logs, VA Scan Logs, Web Server Activity, and Web Cache & Proxy Logs. Switch Logs are also shown connecting different segments.

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Log

Logs are collection of information/data on events generated in the form of an audit trail by the various components of an information system such as network, applications, operating system (OS), service, etc. A log can provide an indication that something may have gone wrong and it helps network defenders in analyzing and detecting issues.

Separately, transaction logs or firewall logs or intrusion prevention system (IPS)/intrusion detection system (IDS) logs do not report faults. They simply store records of specific events; for example, deletion of record from the database. When logs from multiple devices are collected, correlated, and analyzed by security incident and event management (SIEM) systems, something meaningful is generated. For example, by combining the transaction log that represents a record entry by a user with the firewall log that represents network activity from an IP address registered by the same user who made the record entry, we can verify the authenticity of that user.

Logs are recorded and stored through the logging process. Generally, there are four types of logging: security logging, operational logging, compliance logging, and application debug logging. Security logging concentrates on identifying and responding to security-related activities such as threats, viruses, malware, data loss, etc. It records logs about user login, unauthorized access to resources, etc. Operational logging concentrates on system-processing activities. It informs the network defender regarding failures and potentially actionable conditions. It also facilitates service provisioning and financial decisions. Compliance logging is a part of security logging because regulations are developed to enhance the security of systems and data. Application debug logging is logging that is beneficial to application/system developers, not system administrators. It concentrates on recording debugging logs, which are analyzed by the application developer to detect issues. This type of logging can be disabled and enabled in a production system based on circumstantial requirements.

Typical Log Sources

A log source refers to a data source that builds an event log. Almost every device or application on the network has logging capability and can produce a log to record the information regarding an event. Every security system generates logs in some form or another. Windows logs, client and file server logs, router logs, firewall logs, and database logs are examples of the various log sources in the network.

Log sources use two mechanisms to transfer records: pull-based and push-based. In a push-based mechanism, the system or application either saves records on the local disk or sends them over the network. If the records are being sent over the network, then a log collector is needed to collect them. System Logging Protocol (syslog) and Simple Network Management Protocol (SNMP) are the two main push-based protocols through which log records are transferred. In a pull-based mechanism, a system or an application pulls the log records from a log source. It works based on the client-server model. The system or device that follows this mechanism usually stores their log data in a proprietary format. For example, Check Point provides OPSEC C library to pull logs from a Check Point device.

The required log sources need to be configured to collect important information in required formats and locations and store it for a long period of time. Log source configuration is not an easy task. Initially, the hosts and host components that are going to participate in log management infrastructure need to be identified based on the standard rules and policies. A single log file includes information from multiple sources; for example, an OS log includes information not only from OS itself but also from various other security programs. Once the log source is determined, the types of events to be logged by each log source as well as the features of data to be logged for each event need to be specified. Then, the log sources need to be configured based on the features provided by that particular type of log source. Some log sources provide granular configuration options while others provide no granularity at all. In log sources with no granularity, logging is either enabled or disabled, without any control over the kind of data that can be logged.

Need of Logs



- 1 To identify **security incidents**
- 2 To monitor **policy violations**
- 3 To identify **fraudulent activity**
- 4 To identify **operational** and **long-term** problems
- 5 To establish **baselines**
- 6 To ensure compliance with **laws, rules, and regulations**

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Need of Logs

Logs are needed to understand the various events occurring on the network and are used for various purposes—from performance to threat identification. They are a good source of forensic information and help understand "what happened" after the occurrence of a security incident. Each log file comprises a variety of information, among which some may be invaluable. A proper analysis of log data enables actionable information to be identified, which helps the network defender in detecting and monitoring potential security breaches, internal misuse of information, operational issues, and long-term issues. It also helps validate whether the end-user has followed all the documented protocols to detect fraudulent activities and policy violations. It is also useful in internal investigations, security auditing and forensic analysis, determination of operational trends, and implementation of baselines. It provides information about what, how, and why a particular security intrusion occurred and, thus, helps in recovery and mitigating processes as well. It also ensures compliance with laws, rules, and regulations for storing and analyzing log data. Further, it can act as an audit trail for auditing purposes.

Logs can help in the following tasks.


- **System monitoring:** Logs provide detailed information about the transactions that are occurring across the environment. This facilitates constant system monitoring and helps in determining errors, anomalies, and suspicious system activities; it also helps respond to such situations as early as possible.

- **Troubleshooting:** The various information and messages available in log files can be utilized to troubleshoot a problem. However, these log files are not enough to troubleshoot network problems. Syslog need to be utilized for this purpose. It records events and arranges them into log files, which is beneficial in monitoring the various activities of the OS and in troubleshooting issues.
- **Forensics and analysis:** Logs play an important role in forensics and analysis process. They are a permanent source of record that cannot be altered through the normal course of actions. Logs are stored in a chronological sequence, thus they describe not only what happened but also when and how it happened. When logs are sent to another host or a central log collector, they act as a backup source of evidence and are especially useful if the original copy is suspected to have been tampered. If the information on the original source is found suspicious, then the separate copy is considered for authentication.

Logs also support the findings of other evidences and improve their authenticity if their findings corroborate. Often, identifying the complete scenario of an event is not only dependent on one source of information but multiple sources such as files and their corresponding timestamps, network data, logs, etc. Logs may also assist in rejecting other evidences if they are suspected to have been tampered by an attacker.

- **Incident response:** Incident response activity requires proper correlation of log events across all devices and assets. This helps in determining the extent and impact of a network compromise and the steps required for remediation. However, the various security devices in a network may not have the required correlation capabilities to provide a complete picture of the attacker's activities during an attack. A common solution to this problem is to correlate the activities using log files.

Logging Requirements



Before enabling **logging capability**, you should know:

- What to log
- Where to store the logs
- Methods for logging
- Tools required for logging
- Log format

You should be able to:

- Perform regular tuning and review of logs
- Synchronize timestamps of all the sources to perform correlation
- Prevent unauthorized access and manipulation to the logs
- Correlate the data sources to identify any malicious activity
- Analyze the security-based events that are to be stored in the event logs

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Logging Requirements


Before setting the requirements for a logging solution, details such as what to log, where to store the logs, methods for logging, tools required for logging, log format, etc. should be known.

Requirements for Logging

For effective security events logging, you should be able to do the following:

- Determine applications and systems (including those that are outsourced or are on the cloud) on which event logging is enabled
- Configure the information system for providing correct security incidents
- Perform regular tuning and review of logs to minimize the number of false positives
- Store events in event logs
- Normalize and aggregate security-related events
- Correlate the data sources to identify any malicious activity
- Synchronize timestamps of all the sources to perform correlation
- Prevent unauthorized access and manipulation of the logs
- Analyze security-based events that are to be stored in the event logs

Typical Log Format



Log file contains various types of information that helps provide **valuable** and **actionable** information

To identify actionable information from the logs, proper **log analysis** and **monitoring** is required

Typical log includes following types of information:

- User identification information
- Date and time
- Type of event
- Success or failure indication
- Event origination point
- Description
- Severity
- Service name
- Protocol
- User

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Typical Log Format

A log file contains various types of valuable and actionable information. The timestamp is an important item in any log as it tells when an event occurred. Therefore, all systems must be configured to synchronize their time from an authoritative time source. The timestamp also helps in filtering data in search results and in identifying the logs that are not matching the standards and require attention. It can help correlate events from multiple systems and present the complete picture of an event. In addition to a timestamp, a description of the event and why it occurred is also recorded. In a network-based event, the IP address information is also present in a log. Further, authenticated user information should also be available in a log. The information present in a log helps network defenders in monitoring user activities or analyzing them if something goes wrong.

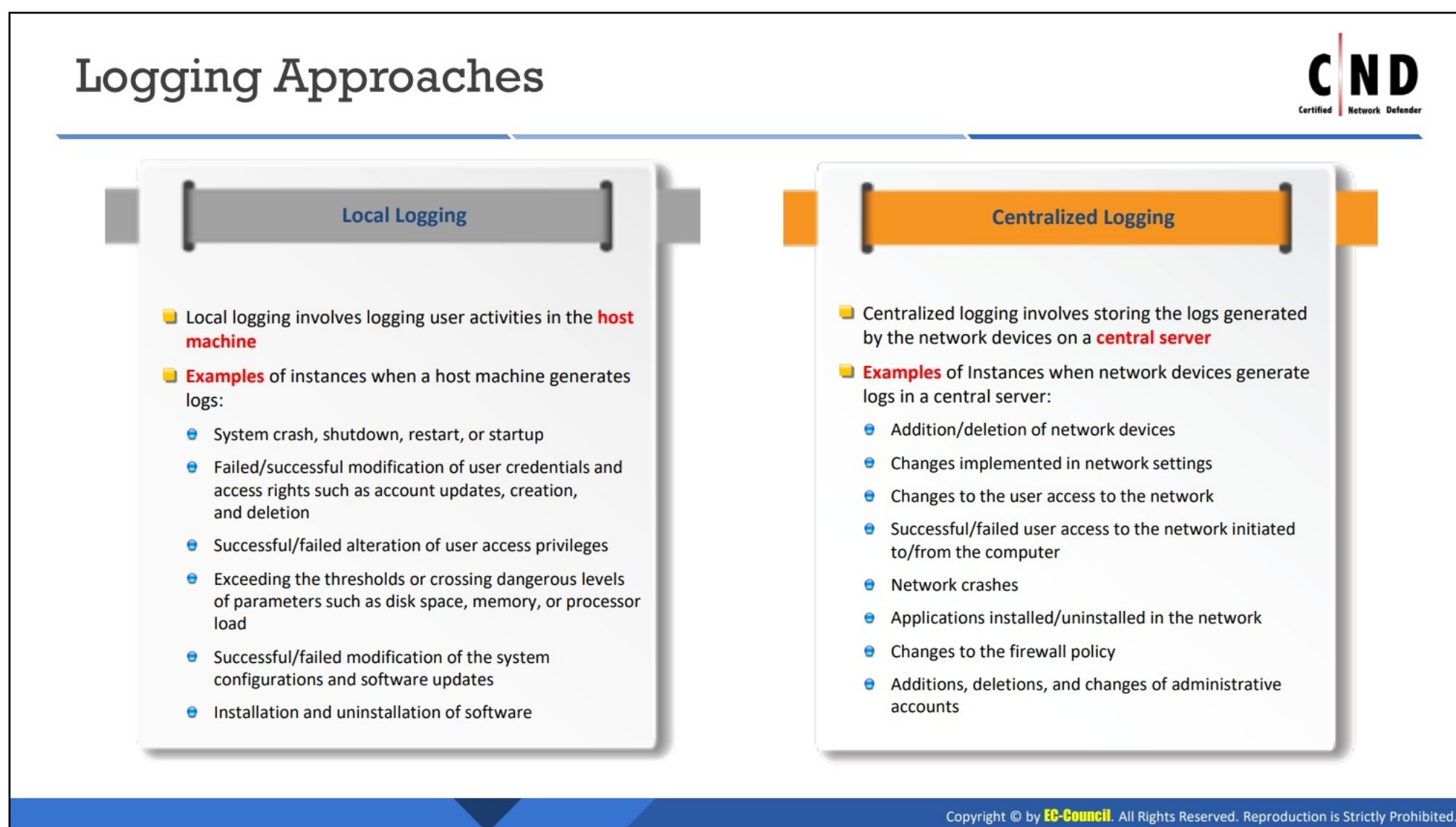
The following are the types of information included in a log:

- User identification information
- Date and time
- Type of event
- Success or failure indication
- Event origination point
- Description
- Severity
- Service name
- Protocol

- User

There are also several things that need attention while setting up the configuration of a log. It should be ensured that private or protected information is not transmitted to the log files. Additionally, information such as passwords, encryption keys, bank information, credit card information, personal identifiable information, personal health information, source code, etc. should not be logged. Only the following should be sent to a log file:

- Initialization/clearing of audit logs
- Creation/deletion of system-level objects
- All access to the log
- All administrative access
- All actions taken by administrators
- All actions taken by a person having root or administrative privileges



Logging Approaches

Logs are stored either on local disks or the central server, depending on the size of the system.

Local logging

Local logging involves logging user activities in the host machine. In other words, it is the process of writing logs into files stored on the local disk. This approach is used by those systems that have a limited number of hosts. If a system has many hosts, then it becomes difficult to manage logs and analyze them. It also becomes complicated to identify security-related events across multiple log files on multiple servers. A common solution to this problem is to use centralized logging.

Example of instances when a host machine generates a log record:

- System crash, shutdown, restart, or startup
- Failed and successful modification of user credentials and access rights such as account updates, creation, and deletion
- Successful/failed alteration of user access privileges
- Exceeding the thresholds or crossing dangerous levels of parameters such as disk space, memory, or processor load
- Successful/failed modification of system configuration and software updates
- Installation and uninstallation of software

■ **Centralized logging**

Centralized logging involves storing the logs generated by the network devices on a central server. In other words, it is the process of collecting and aggregating logs in one central location. It works in four parts: log collection, transport, storage, and analysis.

Benefits of centralized logging:

- Logs stored in a central location are indispensable when trying to troubleshoot security-related problems and determine why they happened.
- It enables proactive management of the network.
- It facilitates in-depth data analysis and delivers greater value.
- It minimizes the risk of losing data.
- It enhances network security.

Examples of instances when a network device generates a log record in a central server:

- Addition/deletion of network devices
- Changes implemented to network settings
- Changes in user access to the network
- Successful/failed user access to the network initiated to/from the computer
- Network crashes
- Applications installed/uninstalled in the network
- Changes to the firewall policy
- Additions, deletions, and changes in administrative accounts



LO#02: Discuss log monitoring and analysis on Windows systems

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

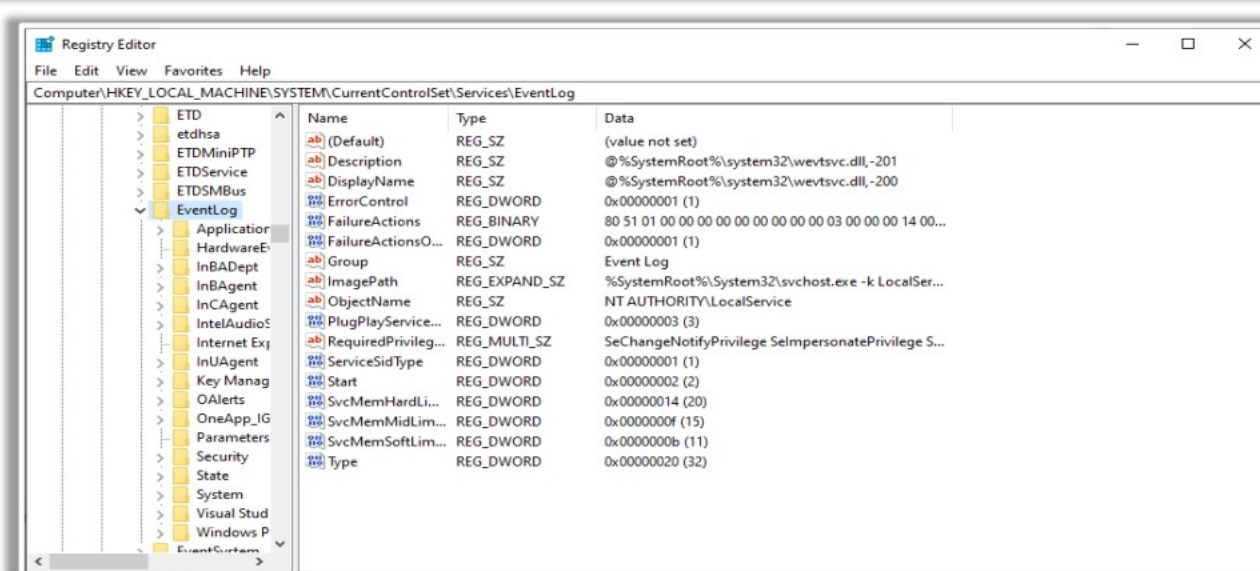
LO#02: Log Monitoring and Analysis on Windows Systems

The objective of this section is to explain monitoring and analysis of logs in Windows systems. It describes Windows event logs, their types, and how to monitor and analyze them.

Windows Logs



- Windows OS tracks various events, activities, and functions through logs
- Windows event logs, consisting of a header and a series of **event records**, provide a standard, centralized way for applications (and the OS) to record important software and hardware events
- Windows Event log audit configurations (i.e., log retention, log size, etc.) are recorded based on the registry key:
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\<Event Log>



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Windows Logs

Windows event logging service collects events from multiple sources and keeps them in a single location known as Windows event log. These logs act as the primary source of evidence for all important actions/activities on a Windows system. Windows event log contains logs of system, security, and application notifications that are monitored and analyzed by network defenders to detect issues in the system. It provides a standard, centralized way for applications (and the OS) to record important software and hardware events. It uses a structured data format that simplifies the process of searching and filtering for a particular type of log. Windows event log files can be viewed through Event Viewer, which is the programming interface that facilitates analysis of these logs. Each event is a log entry that includes information such as event time, event source that caused the event, event type (Information, Warning, Error, Success Audit, or Failure Audit), and event ID for the event type.

Windows event log audit configurations, that is, log retention, log size, etc. are recorded based on the below registry key.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\<Event Log>

This key comprises various subkeys, which are known as logs. Each log includes registry values such as CustomSD, DisplayNameFile, DisplayNameID, File, MaxSize, etc., which can be configured as per requirement.

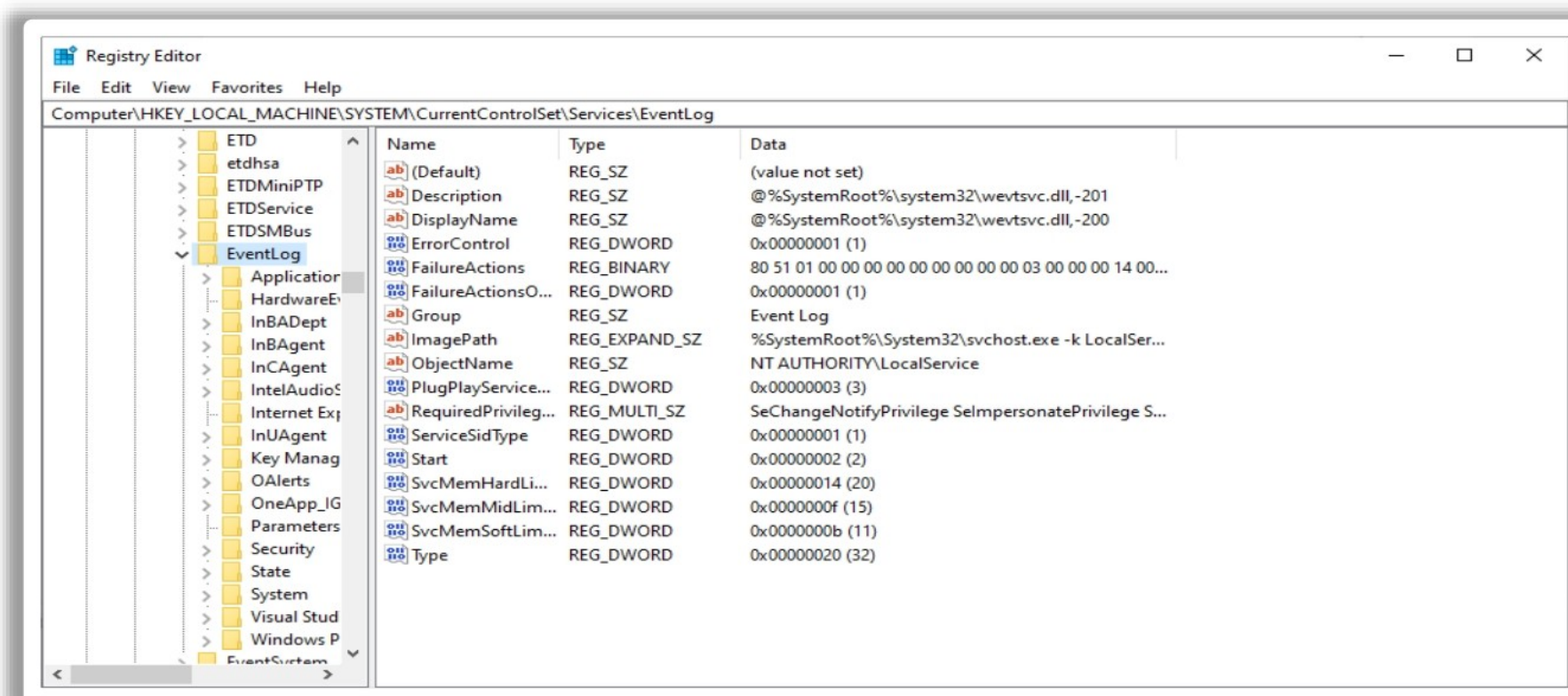


Figure 15.1: Screenshot of Registry Editor

Windows Event Log File Internals

In simple terms, Windows event log files are databases with records related to the system, security, and applications. The databases related to the system are stored in a file named **System.evtx**, the databases related to security are stored in a file named **Security.evtx**, and the databases related to applications are stored in a file named **Application.evtx**. These Windows event log files are stored in **C:\Windows\System32\winevt\Logs** folder, as shown in the below figure:

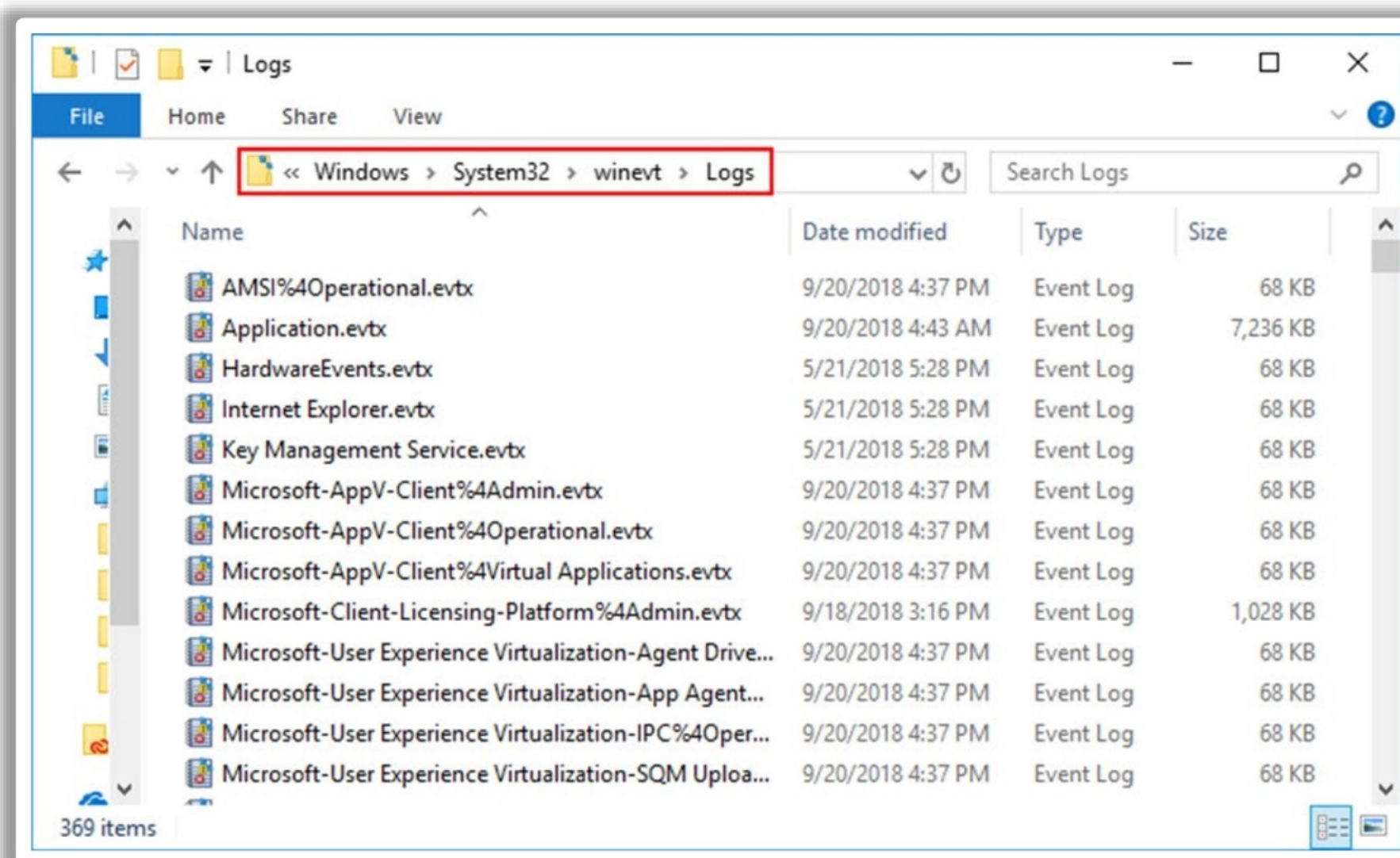


Figure 15.2: Screenshot of Windows event log files

All **.evtx** files can be opened and read with Event Viewer.

Event Log File Format

Each event log consists of a header of fixed size (represented by the `ELF_LOGFILE_HEADER` structure), a variable number of event records (represented by `EVENTLOGRECORD` structures), and end-of-file record (represented by the `ELF_EOF_RECORD` structure).

When the event log is created and updated, both the `ELF_LOGFILE_HEADER` structure and the `ELF_EOF_RECORD` structure are written to it.

When an application calls the `ReportEvent` function to write an entry to the log file, the system then passes the parameters to the event-logging service, which uses the information to write an `EVENTLOGRECORD` structure to the event log file. This is represented in the below diagram:

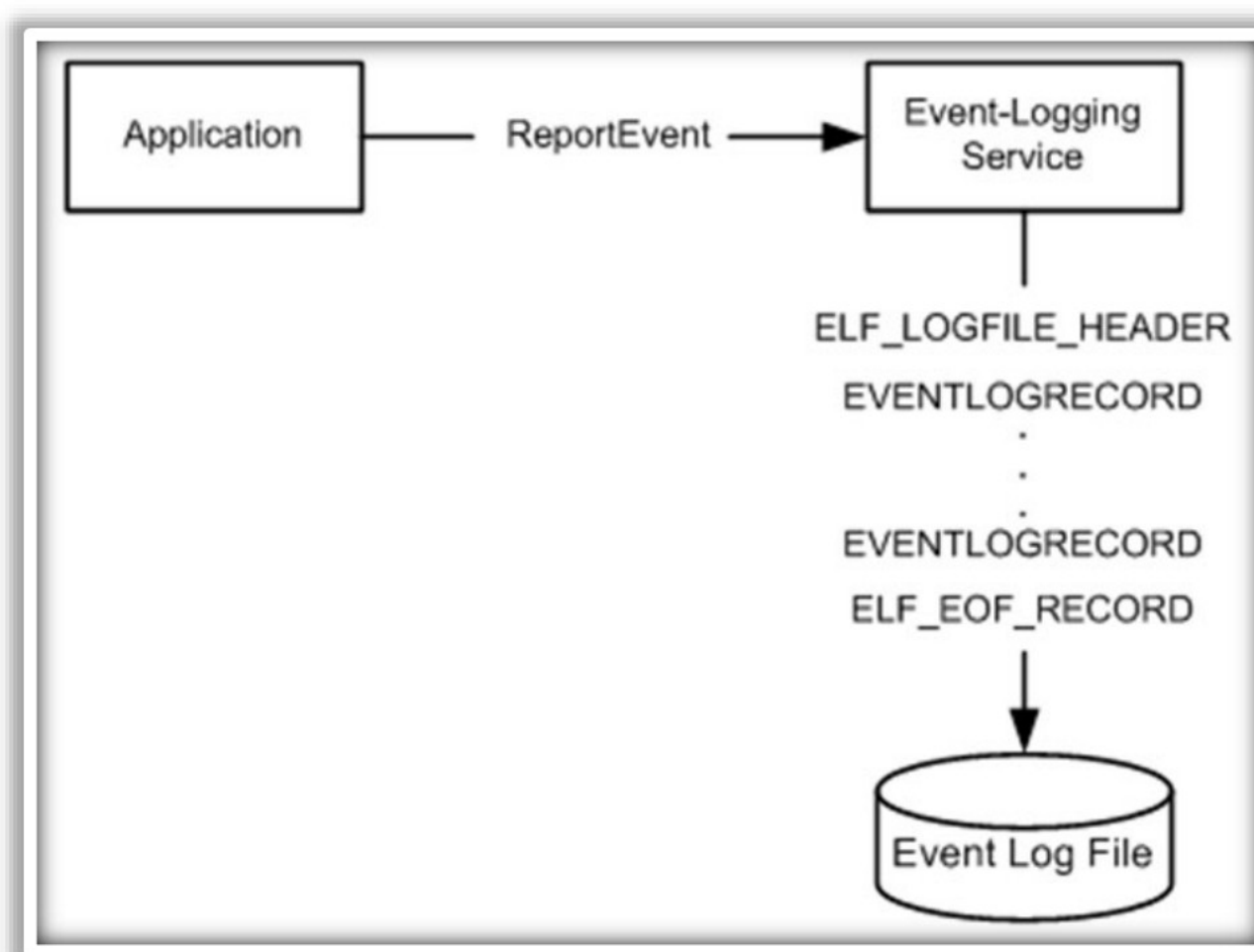


Figure 15.3: ReportEvent Function

There are two ways of arranging the event records, as listed below.

- **Nonwrapping method:** In this method, the oldest record is inserted just after the event log header and new records are inserted just before the `ELF_EOF_RECORD`. In the below example, event records are organized as per the nonwrapping method:

HEADER	(<code>ELF_LOGFILE_HEADER</code>)
EVENT RECORD 1	(<code>EVENTLOGRECORD</code>)
EVENT RECORD 2	(<code>EVENTLOGRECORD</code>)
EOF RECORD	(<code>ELF_EOF_RECORD</code>)

The nonwrapping method can be applied every time an event log is generated or deleted. The event log records continue to organize as per the nonwrapping method until its size reaches the maximum limit. The event log size depends either upon the `MaxSize` configuration value or the number of system resources. After the event log size reaches to its limit, it starts using the wrapping method.

- **Wrapping method:** In this method, event logs are arranged in the form of a circular buffer, in which the oldest event logs are replaced by the new event log. Consider the below example:

```
HEADER                                (ELF_LOGFILE_HEADER)
Part of EVENT RECORD 300 (EVENTLOGRECORD)
EVENT RECORD 301                    (EVENTLOGRECORD)
.
.
.
EVENT RECORD 400                    (EVENTLOGRECORD)
EOF RECORD                          (ELF_EOF_RECORD)
Wasted space
EVENT RECORD 102                    (EVENTLOGRECORD)
EVENT RECORD 103                    (EVENTLOGRECORD)
.
.
.
EVENT RECORD 299                    (EVENTLOGRECORD)
Part of EVENT RECORD 300 (EVENTLOGRECORD)
```

In the above example, event record 102 is the oldest record instead of 1 as the oldest event records from 1 to 101 have been replaced with the newest ones. Suppose if the size of the newest event record is 100 bytes long and the two oldest records are of 65 bytes each, then the system will erase both the two oldest records. The remaining 30 bytes will be utilized later when new event logs occur.

In this method, the size of an event log file is fixed. When the event records in the log file are overwritten, the record at the end of the file is divided into two records. For example, if the size of the record is 200 bytes and the space before the end of the file is 100 bytes only, then the first 100 bytes will be recorded at the end of the file and the other 100 bytes will be recorded just after the ELF_LOGFILE_HEADER. If the existing space at the end of the file is less than the fixed size of EVENTLOGRECORD, then all the new event log records will be recorded just after the ELF_LOGFILE_HEADER and the unutilized space at the end of the file will be occupied by the 0x00000027 pattern.

ELF_LOGFILE_HEADER Structure:

The event-logging service adds the ELF_LOGFILE_HEADER at the start of the event log, which describes information about the event log. Its syntax is described below:

```
typedef struct _EVENTLOGHEADER {
    ULONG HeaderSize;
```



```

    ULONG Signature;
    ULONG MajorVersion;
    ULONG MinorVersion;
    ULONG StartOffset;
    ULONG EndOffset;
    ULONG CurrentRecordNumber;
    ULONG OldestRecordNumber;
    ULONG MaxSize;
    ULONG Flags;
    ULONG Retention;
    ULONG EndHeaderSize;
} EVENTLOGHEADER, *PEVENTLOGHEADER;

```

The following table shows the various members of ELF_LOGFILE_HEADER:

Members	Description
HeaderSize	The size of the header structure, which is always 0 × 30
Signature	The signature is always 0x654c664c, which is ASCII for eLfl
MajorVersion	The major version number of the event log and is always set to 1
MinorVersion	The minor version number of the event log and is always set to 1
StartOffset	The offset to the oldest record in the event log
EndOffset	The offset to the ELF_EOF_RECORD in the event log
CurrentRecordNumber	The number of the next record that will be added to the event log
OldestRecordNumber	The number of the oldest record in the event log. Its value is set to 0 for an empty file
MaxSize	The maximum size, in bytes, of the event log. It is defined when the event log is created
Flags	The status of the event log. It could be one of the four values: ELF_LOGFILE_HEADER_DIRTY0x0001, ELF_LOGFILE_HEADER_WRAP0x0002, ELF_LOGFILE_LOGFULL_WRITTEN0x0004, and ELF_LOGFILE_ARCHIVE_SET0x0008
Retention	The retention value of the file when it is created

EndHeaderSize	The signature is always 0x654c664c, which is ASCII for eLfl
----------------------	---

Table 15.1: Various members of ELF_LOGFILE_HEADER

Flags

Values	Meaning
ELF_LOGFILE_HEADER_DIRTY0x0001	Indicates that records have been written to an event log, but the event log file has not been properly closed
ELF_LOGFILE_HEADER_WRAP0x0002	Indicates that records in the event log are wrapped
ELF_LOGFILE_LOGFULL_WRITTEN0x0004	Indicates that the most recent write attempt failed due to insufficient space
ELF_LOGFILE_ARCHIVE_SET0x0008	Indicates that the archive attribute has been set for the file. Normal file APIs can also be used to determine the value of this flag

Table 15.2: Various values of flag

EVENTLOGRECORD Structure

The EVENTLOGRECORD structure contains information on a single event. Its syntax is described below:

```
typedef struct _EVENTLOGRECORD {
    DWORD Length;
    DWORD Reserved;
    DWORD RecordNumber;
    DWORD TimeGenerated;
    DWORD TimeWritten;
    DWORD EventID;
    WORD EventType;
    WORD NumStrings;
    WORD EventCategory;
    WORD ReservedFlags;
    DWORD ClosingRecordNumber;
    DWORD StringOffset;
    DWORD UserSidLength;
    DWORD UserSidOffset;
}
```



```

DWORD DataLength;
DWORD DataOffset;
} EVENTLOGRECORD, *PEVENTLOGRECORD;

```

The following table represents the various members of EVENTLOGRECORD structure:

Component	Size	Description
Length	4 bytes	Size in bytes of the structure
Reserved	4 bytes	Serves as a signature for the structure
RecordNumber	4 bytes	It is mapped directly from the record ID
TimeGenerated	4 bytes	Time when the event was generated
TimeWritten	4 bytes	Time when the event was written
EventID	4 bytes	EventID generated by the event source
EventType	2 bytes	Type of event
NumStrings	2 bytes	Number of strings in the Strings field. Its value must be between 1 and 256
EventCategory	2 bytes	Event category
ReservedFlags	2 bytes	Specifies whether or not the last string in the Strings field contains well-formed XML. The value 0 × 0000 indicates that the event does not contain XML and the value 0 × 8000 indicates that the event contains XML
ClosingRecordNumber	4 bytes	MUST be set to zero when sent and MUST be ignored on receipt
StringOffset	4 bytes	This MUST be the offset in bytes from the beginning of the structure to the Strings field. If the Strings field is not present (NumStrings is zero), this can be set to any arbitrary value when sent and MUST be ignored on receipt by the client
UserSidLength	4 bytes	Size in bytes of the user's security identifier, which is located within the UserSid field. If there is no UserSid field for this event, this field MUST be set to zero
UserSidOffset	4 bytes	This MUST be the offset in bytes from the beginning of the structure to the UserSid field. If the UserSid field is not present (i.e.,

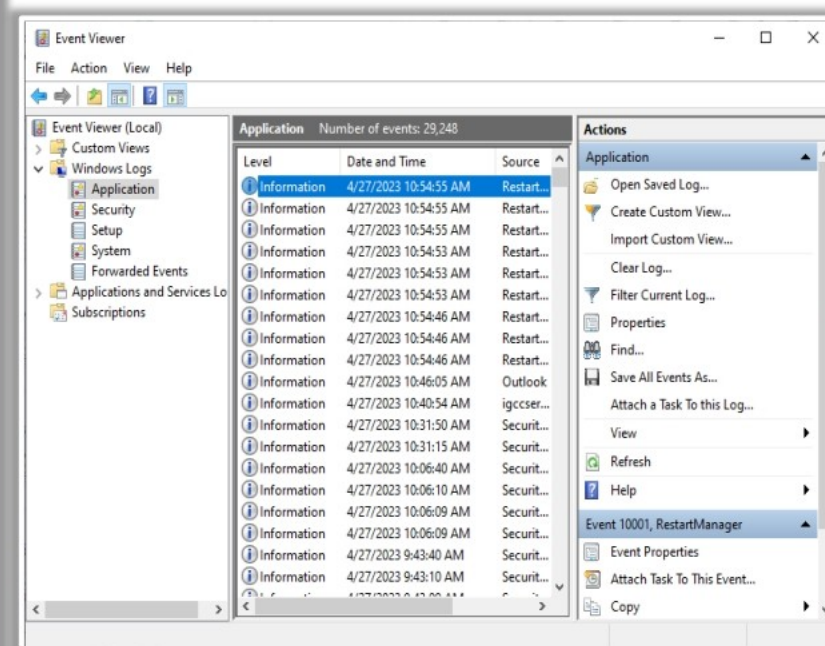
		if UserSidLength is zero), this can be set to any arbitrary value when sent and MUST be ignored by on receipt the client
DataLength	4 bytes	This MUST be the size in bytes of the Data field. If the Data field is not used, this field MUST be set to zero
DataOffset	4 bytes	This MUST be the offset in bytes from the beginning of the structure to the Data field. If the Data field is not present (i.e., if DataLength is zero), this can be set to any arbitrary value when sent and MUST be ignored on receipt

Table 15.3: Various Members of EVENTLOGRECORD Structure

Windows Event Log Types and Entries



- **Event Viewer** provides a quick overview of when, where, and how an event occurred
- Check **Windows Event Log** for various types of logs:
 - **System logs:** Windows and Windows system service logs
 - **Security logs:** Audit logs based on success/failed events
 - **Setup logs:** Configuration logs
 - **Application logs:** Events based on severity categorized
 - **Forwarded event logs:** Events forwarded by another computer in a network
- Typical log entries contain the following types of information about the events:
 - **Level:** It defines the **severity of the event**; various types of severity levels are Error, Warning, Information, Success Audit, and Failure Audit
 - **Keywords:** It defines the **type of event** occurred; various types of events are AuditFailure, AuditSuccess, Classic, Correlation Hint, Response Time, SQM, WDI Context, and WDI Diag
 - **Date and Time:** It defines the **date of events** occurred
 - **Source:** It defines the **source of the event**
 - **Event ID:** A **unique** event ID is assigned for each type of event
 - **Task Category:** It defines task categories



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Windows Event Log Types and Entries

Windows Event Viewer is a tool that provides a quick overview of when, where, and how an event occurred. It provides detailed information about events, errors, warnings, and information messages that are generated by the OS. It is especially useful for troubleshooting issues.

In Event Viewer, events are stored in Windows logs files under different categories such as application, security, system, setup, and forwarded.

- **Application event log:** This includes events related to the applications installed on the system; specifically, informational events, warnings from the applications, and errors raised in an application. For example, if an application such as Microsoft Excel breaks down, then this event will be logged into Windows event log with the name of the application and why it is crashed.
- **Security event log:** This includes events related to Windows security; specifically, log-on/log-off activities, resource access, and also information based on Windows system's audit policies. It is analyzed by network defender to identify attempted and/or successful unauthorized activities. For example, if the system attempts to verify account credentials when an end-user tries to log-on to a machine.
- **Setup event log:** This includes enterprise-focused events that cover all actions that occurred during installation; for example, the location of memory dump from bug checks.
- **System event log:** This includes events that are logged by the OS segments; specifically, information about hardware changes, system changes, device drivers, and all machine-related activities (for example, failure of the device driver).

- **Forwarded event log:** This includes events that are received from other systems present on the same network.
- **Custom log:** A custom log facilitates an application to change the size of the log or add access control lists (ACLs) without influencing other applications.

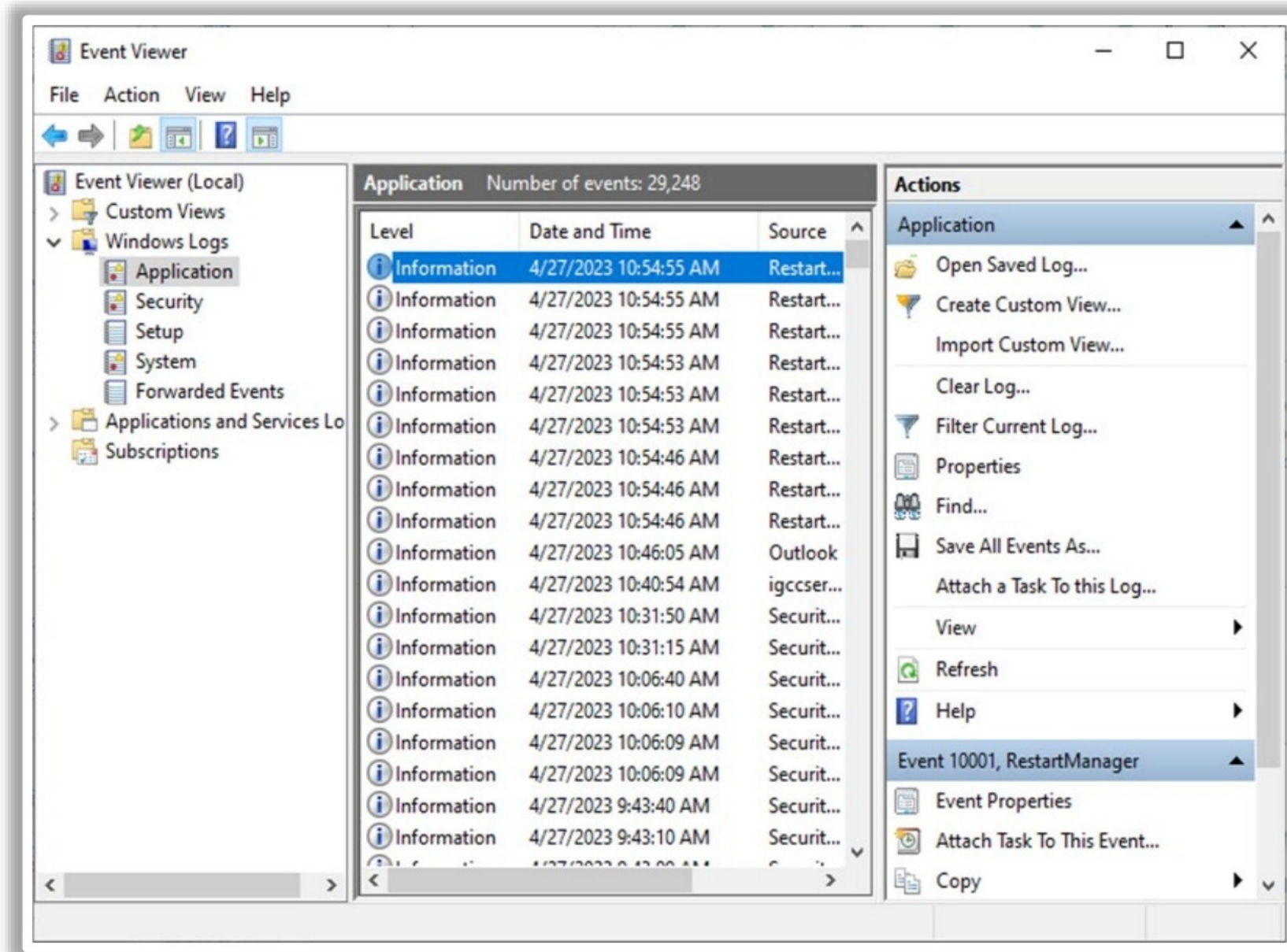


Figure 15.4: Screenshot of Windows Event Viewer

Each log entry in a particular log type contains the following types of information about the event:

- **Level:** It defines the severity of the event. The various types of severity levels are Error, Warning, Information, Success Audit, and Failure Audit.
- **Keywords:** It is a set of categories or tags that defines a type of event. The various types of events are AuditFailure, AuditSuccess, Classic, Correlation Hint, Response Time, SQM, WDI Context, and WDI Diag.
- **Date and time:** It defines the date on which an event occurred.
- **Source:** It defines the source of the event.
- **Event ID:** A unique event ID is assigned for each type of event.
- **Task category:** It defines the category of task.
- **User:** It defines the username on whose behalf a particular event was generated.
- **Operational code:** It defines the activity that an event was performing when an event was raised.
- **Log:** It defines the name of the log on which the event was recorded.

- **Computer:** It defines the computer name on which the event was raised.

Additional event properties can be viewed by adding columns in Event Viewer display. To do so, click on View (menu bar) and then Add/Remove Columns. The following types of properties can be added or removed.

- **Process ID:** It defines the process identification number for the generated event.
- **Thread ID:** It defines the thread identification number for the generated event.
- **Processor ID:** It defines the processor identification number that processed the event.
- **Session ID:** It defines the terminal server session identification number in which the event was raised.
- **Kernel time:** It defines the time taken in executing kernel-mode instructions in CPU time units.
- **User time:** It defines the time taken in executing user-mode instructions in CPU time units.
- **Processor time:** It defines the time taken in executing kernel-mode instructions in CPU ticks.
- **Correlation ID:** It defines the activity in the process for which the event was involved.
- **Relative correlation ID:** It defines the related activity in a process for which the event was involved.

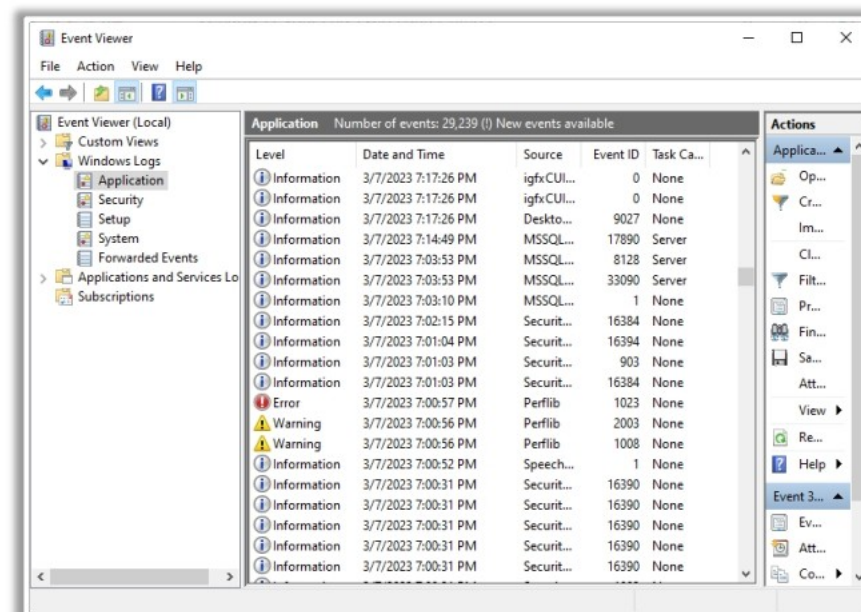
You can also view XML representation of an event by clicking the Details tab in an event's properties.

Event Types



Event Type	Description
Error	An event indicates a significant problem such as loss of data or loss of functionality. For example, if a service fails to load during startup, an Error event is logged
Warning	An event that is not necessarily significant but may indicate a possible future problem . For example, when disk space is low, a Warning event is logged. If an application can recover from an event without loss of functionality or data, it can generally classify the event as a Warning event
Information	An event that describes successful operation of an application, driver, or service. For example, when a network driver loads successfully, it may be appropriate to log an Information event. Note that it is generally inappropriate for a desktop application to log an event each time it starts
Success Audit	An event that records an audited security access attempt that is successful . For example, a user's successful attempt to log on to the system is logged as a Success Audit event
Failure Audit	An event that records an audited security access attempt that fails . For example, if a user tries to access a network drive and fails, the attempt is logged as a Failure Audit event

Source: <https://docs.microsoft.com>



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Event Types

Events are categorized into five types, based on their severity levels.

- **Error:** This type of event describes a significant problem such as loss of data or functionality. For example, an error event is recorded when a service is unable to load at startup.
- **Warning:** This type of event is of less importance but may describe a possible future problem. For example, a warning event is recorded when there is low space on the disk. Events are also classified as a warning event when an application can recover from an event without any loss.
- **Information:** This type of event indicates the successful operation of an application, driver, or service. For example, an information event is recorded when an application driver loads successfully.
- **Success audit:** This type of event is recorded when any successfully audited security access attempt is detected. For example, a success audit event is recorded when a user successfully logs on to the system.
- **Failure Audit:** This type of event is recorded when any unsuccessful audited security access attempt is detected. For example, a Failure Audit event is recorded when a user fails in accessing a network drive.

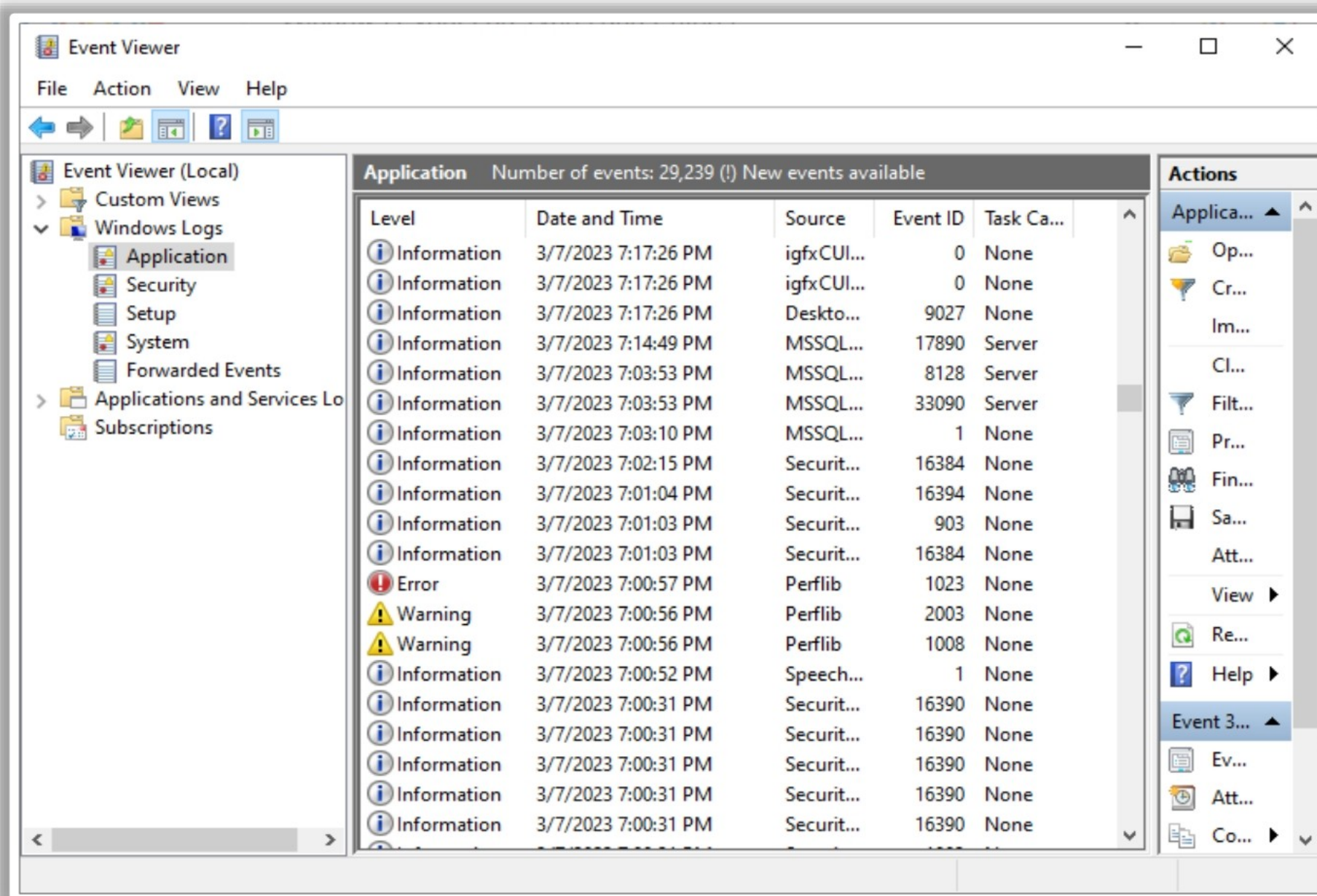
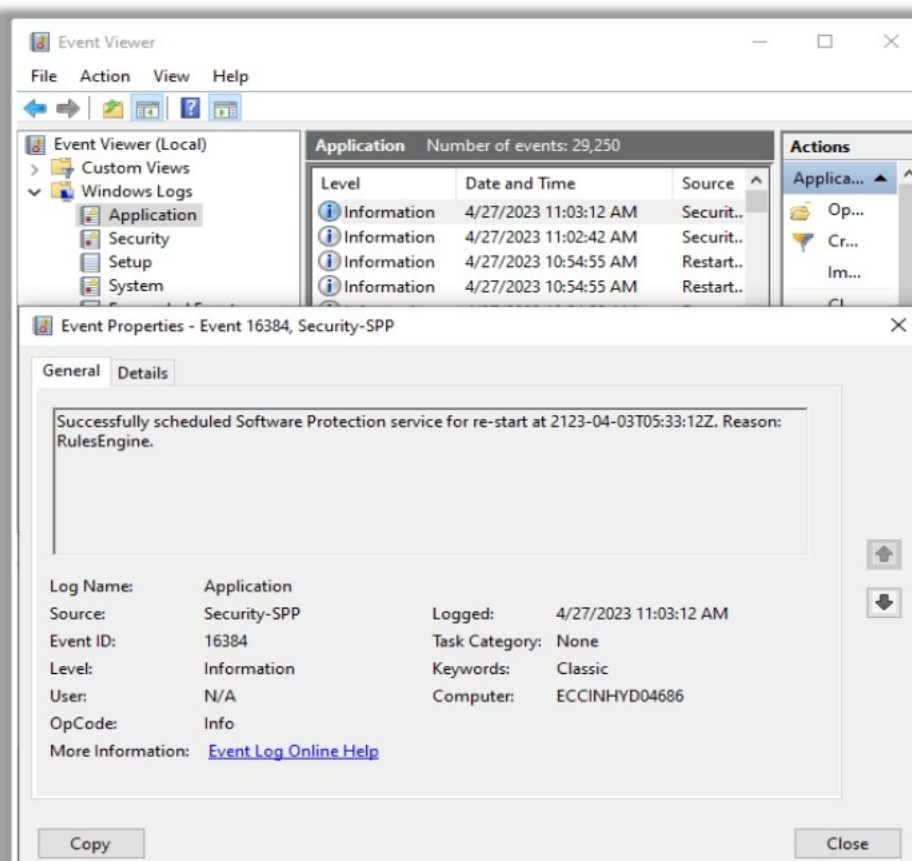


Figure 15.5: Various Windows Event Types

Monitoring and Analysis of Windows Logs



- Open **Event Viewer**, click the required log you want to view
- In the details pane, click the event that you want to view. Description and header information is displayed in the **Preview Pane**
- The information displayed in the **Preview Pane** about the event is as follows:
 - Log Name:** The type of Windows log
 - Source:** Source is the cause that is responsible for the event raised by either an individual, or a system, or a program
 - Event ID:** The type of event that occurred
 - Level:** Event level type is divided into five types: Error, Warning, Information, Success Audit, and Failure Audit
 - User:** User responsible and who logged on the computer at the instance of the event
 - Logged:** The timestamp of the event
 - Task Category:** Primarily used in case of security log, which classifies an event based on the event source
 - Computer:** The name assigned to the computer where the event occurred



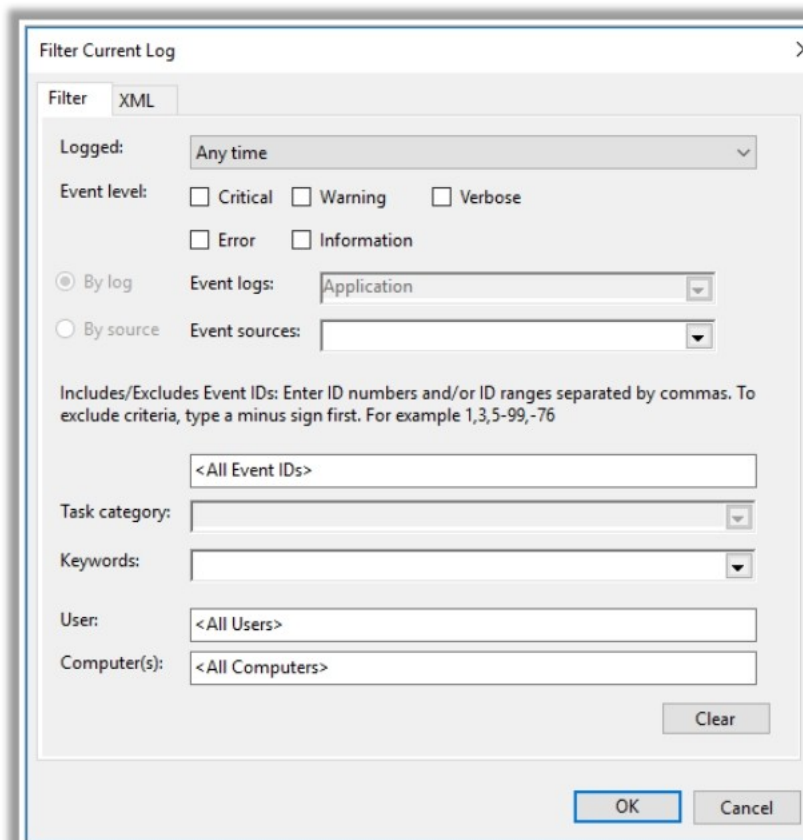
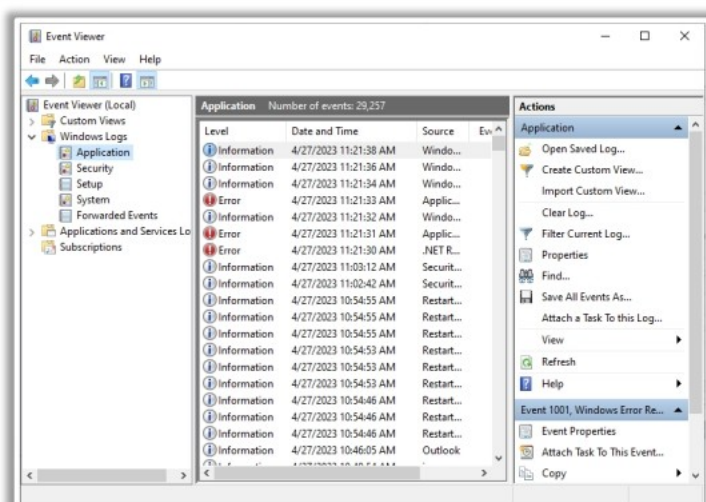
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Windows Logs (Cont'd)



Finding Events in a Log

- The **Filter** feature in Event Viewer allows the removal of clutter from the event log display
- Each log can be independently configured with different filter properties
- Use **Filter** and **Find** features in Event Viewer, under the **Actions** pane
- After applying the filter, the Event Viewer shows the log with matching properties



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Windows Logs (Cont'd)



Examining Event Log Entries

System Log Entries

- The system log contains events **logged by Windows system components**
- System log includes:
 - Changes to the OS
 - Changes to the hardware configuration
 - Device driver installation
 - Service pack update/installation
 - Software and hardware installations
 - Starting and stopping of services
 - System shutdown/restart
 - Log-on failures
 - Alteration of machine information
 - Printing jobs

Application Log Entries

- The application log contains events **logged by applications or programs**
- Application log includes:
 - Installation and removal of a particular software package
 - Confirmation/refutation of virus infection
 - Startup and shutdown of firewall
 - Detection of hacking attempts

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Windows Logs (Cont'd)



Security Log Entries

- The security log is the **mother of all logs** in forensic terms
- Log-ons, log-offs, attempted connections, and policy changes are all reflected in the event contained therein
- Unfortunately, security logging is turned off by default
- It needs to be enabled by the group or local policy to be useful

To support later investigations, enabling local (or group) policy for **audit policy** is recommended with some of the following **actions** at the minimum:

Audit account log-on events	Success, Failure
Audit account management	Success, Failure
Audit log-on events	Success, Failure
Audit policy change	Success, Failure
Audit privilege use	Success, Failure

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Windows Logs

Windows event logs include critical information such as log-on failures, log tampering, failed attempts to access files, etc. They also warn regarding upcoming system issues and protect the system from unexpected disasters. In addition to this, these event logs may also describe an attempt made by a user to compromise the system or an unsanctioned configuration change. Thus, these event logs need to be monitored and analyzed to identify network vulnerabilities, security breaches, and threat intruders. These event logs enable network defender to protect the

network against internal threats and vulnerabilities. The most common way to monitor and analyze Windows event logs is to use the Windows Event Viewer.

Viewing Events in Event Viewer

- Open Windows Event Viewer by clicking the Start icon and then typing "Event Viewer" in the search box.

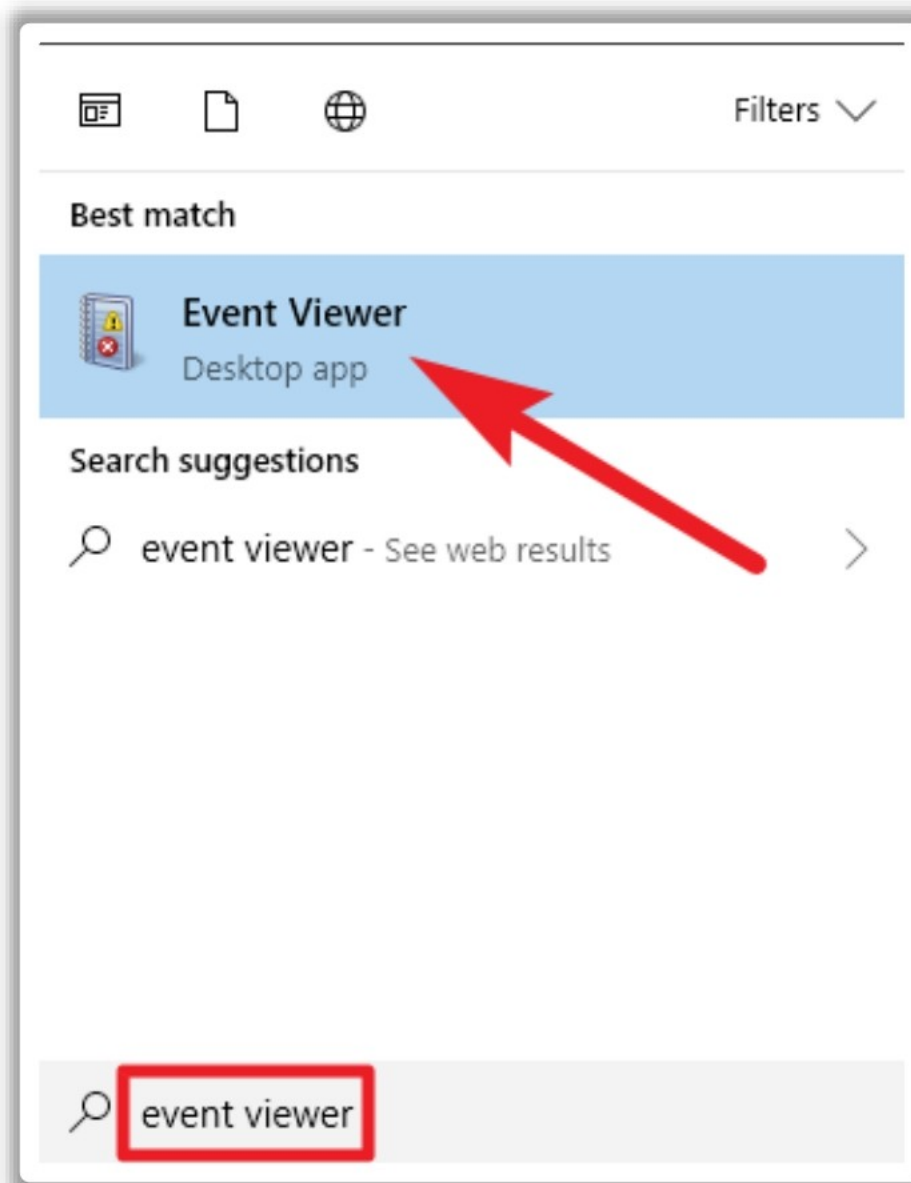


Figure 15.6: Screenshot of the Search Box

- Once Event Viewer opens, click on the required log file from the console tree. A list of events can be seen in the details pane.
- In the details pane, clicking on any specific event will reveal its description and header information in the Preview pane.

The information displayed in the Preview pane about the event is described below.

- **Log name:** The type of Windows log
- **Source:** Source is the cause that is responsible for the event raised by either an individual or a system or a program
- **Event ID:** The type of event that occurred
- **Level:** Event level type is divided into five types: Error, Warning, Information, Success Audit, and Failure Audit
- **User:** User responsible and who logged on the computer at the instance of the event
- **Logged:** The timestamp of the event

- **Task category:** Primarily used in case of a security log that classifies an event based on the event source
- **Computer:** The name assigned to the computer where the event occurred

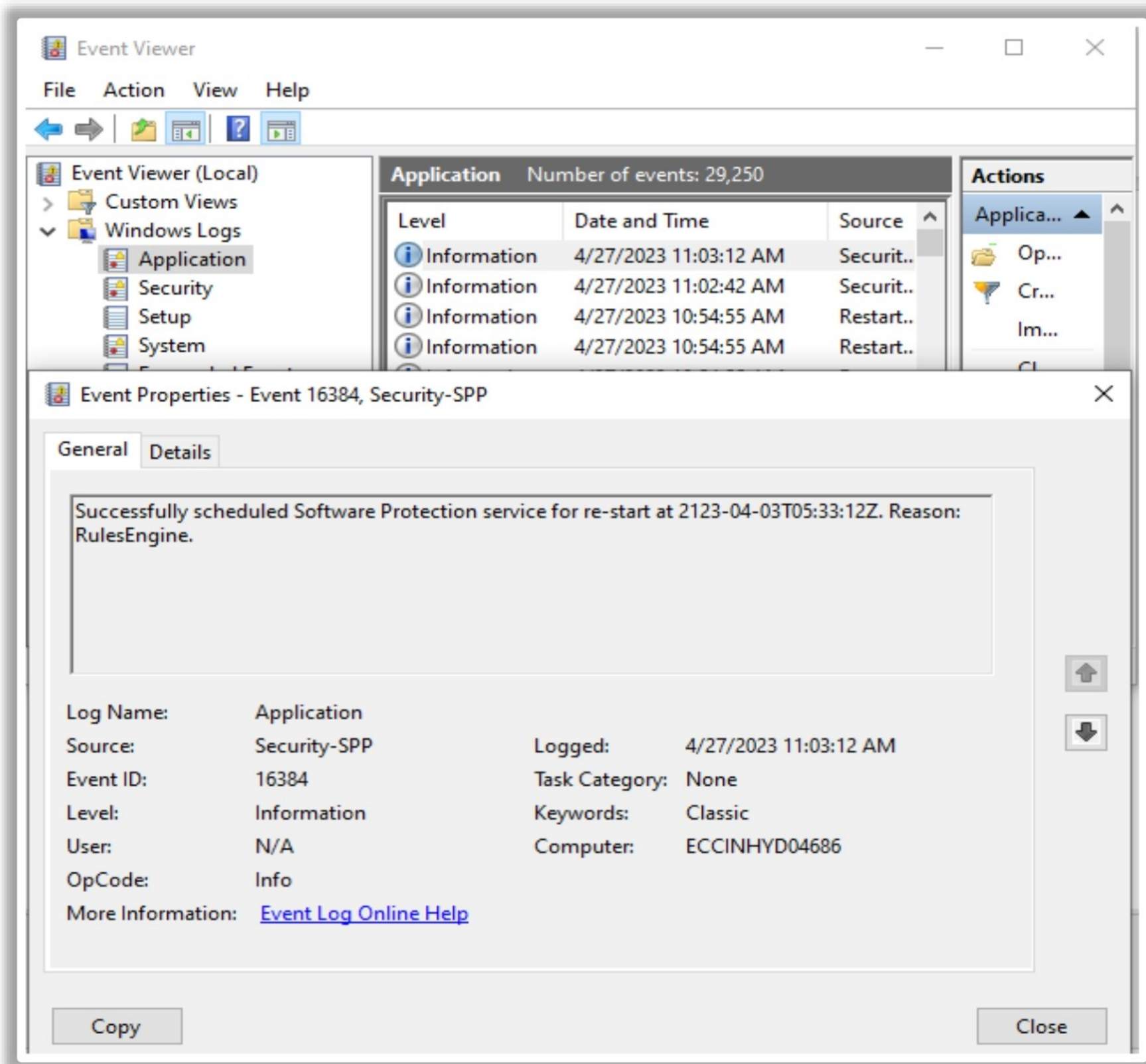


Figure 15.7: Screenshot of Event Viewer

Filtering/Finding Events in Event Viewer

The Filter feature in Event Viewer helps in targeting the information that may be required for investigation. To save time and effort, Event Viewer provides the option to save specific filters for future use through the Create Custom View feature.

Filter feature can allow the removal of clutter from the event log display and limit the data displayed in a single log. Each log can be independently configured with different filter properties.

The following steps are used to create a filter:

- Select the log that needs to be filtered.
- After that, click on "Filter Current Log" option available under the Action pane.
- The "Filter Current Log" dialog box will appear.

- Specify a time period, if the approximate time when the events occurred is known.
- The event levels can be specified from the available options (Critical, Warning, Verbose, Error, and Information). If no option is specified, all event levels will be returned.
- Specific event IDs can be mentioned in the defined format.
- Specific event sources can be selected; similarly, specific keywords, users, or computers can be searched.
- Click OK to close the "Filter Current log" dialog box.
- After applying the filter, the Event Viewer will show the log with matching properties.

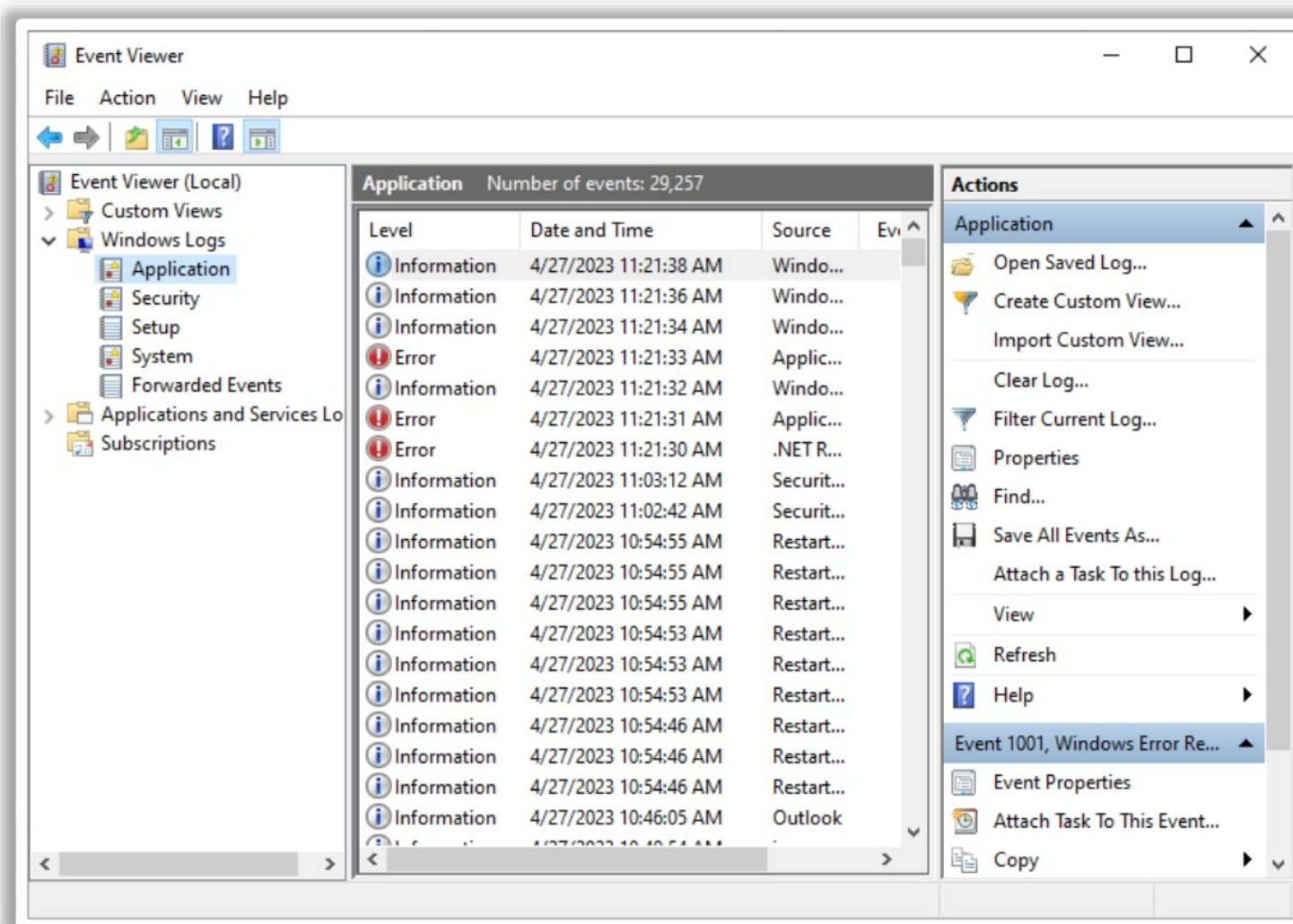


Figure 15.8: Screenshot of Event Viewer

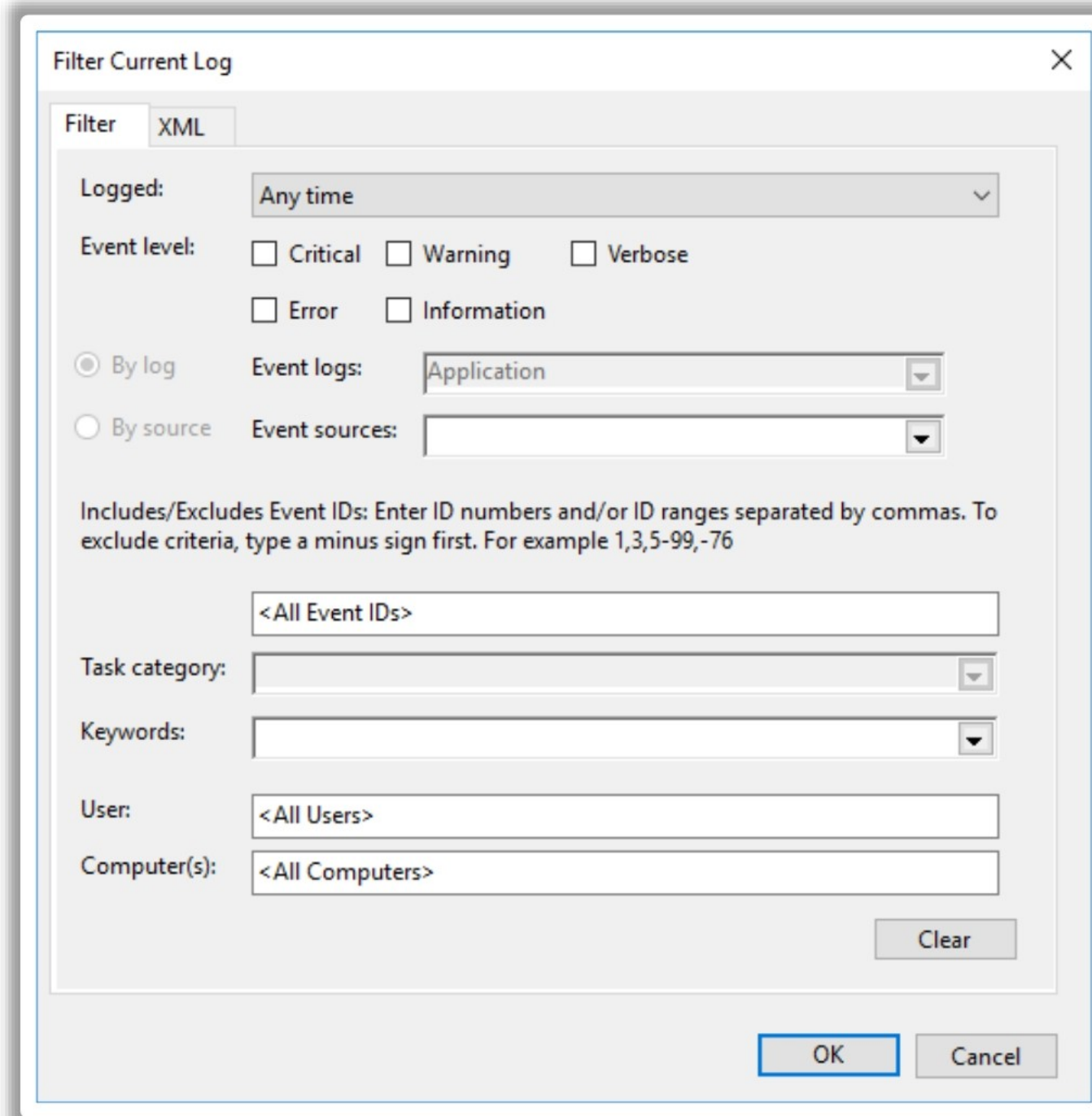


Figure 15.9: Screenshot of "Filter Current Log" Dialog Box

To find an event, follow the steps below:

- Click on "Find" option available under the Action pane
- Type the information that needs to be found and then click Find Next
- Click Close, when search is complete

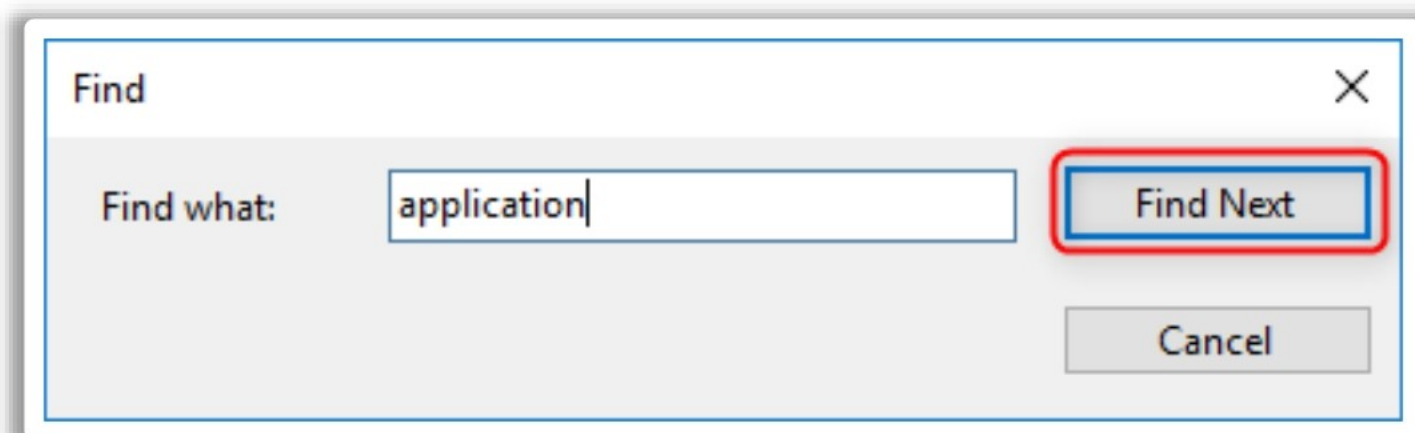


Figure 15.10: Screenshot of Find dialog box

Examining Event Log Entries

Event Viewer displays three types of event log entries, as described below.

■ System log entries

The system log contains events logged by Windows system components. It contains information about system changes such as device driver installations, etc. To view system log entries in Event Viewer:

- Open Event Viewer and then select System log from Windows logs section in the console tree
- A list of system events appears in the details pane
- Select the specific event whose details needs to be viewed

Examples of system log records:

- Changes to the OS
- Changes to the hardware configuration
- Device driver installation
- Service pack update/installation
- Software and hardware installations
- Start/stop of services
- System shutdown/restart
- Log-on failures
- Alteration of machine information
- Printing jobs

■ Application log entries

The application log contains events logged by applications or programs. To view application log entries in Event Viewer:

- Open Event Viewer and then select Application log from Windows logs section in the console tree
- A list of application events will appear in the details pane
- Select the specific event whose details need to be viewed

Examples of application log records:

- Installation and removal of a particular software package
- Confirmation/refutation of virus infection
- Startup and shutdown of firewall
- Detection of hacking attempts

▪ Security log entries

The security log is the mother of all logs in forensic terms. Unfortunately, security logging is turned off by default. To view security log entries in Event Viewer,

- Open Event Viewer and then select Security log from Windows logs section in the console tree
- A list of security events appear in the details pane
- Select the specific event whose details need to be viewed

Examples of security log records:

- Log-ons
- Log-offs
- Attempted connections
- Policy changes

To support later investigations, enabling local (or group) policy for audit policy is recommended with some of the following actions at the minimum:

Audit account log-on events	Success, Failure
Audit account management	Success, Failure
Audit log-on events	Success, Failure
Audit policy change	Success, Failure
Audit privilege use	Success, Failure

Table 15.4: Actions to Enable Local (or Group) Policy for an Audit Policy



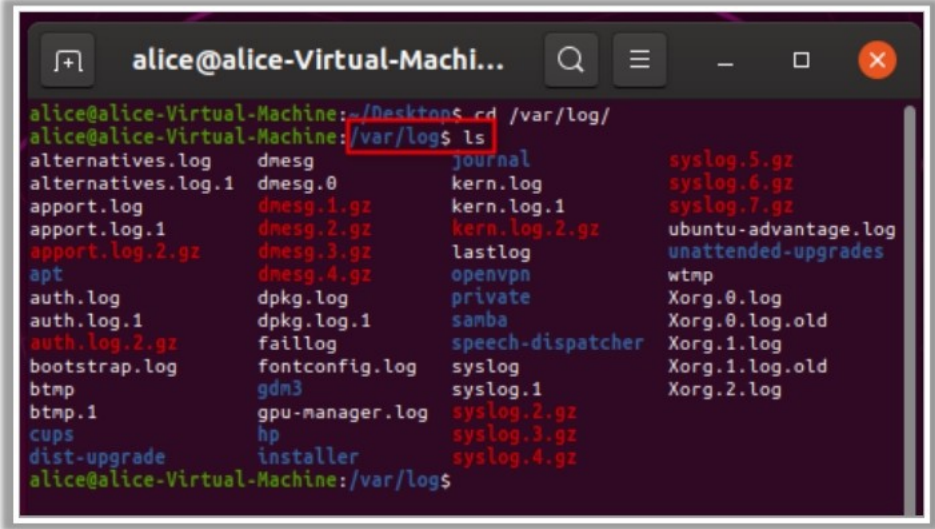
LO#03: Discuss log monitoring and analysis on Linux systems

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

LO#03: Log Monitoring and Analysis on Linux Systems

The objective of this section is to explain monitoring and analysis of logs in Linux-based systems. It describes Linux logs, the various Linux log files, and commands to monitor and analyze Linux logs.

Linux Logs



- Linux logs are a **record of any activity** or **event** in Linux OS
- Most Linux logs are located at **/var/log directory** in plain ASCII text format
- System log daemon (syslogd) produces logs for the system and different programs in Linux OS

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.



























Linux Logs

Linux logs are a record of any activity or event in a Linux-based OS (hereinafter “Linux OS”); they include messages on just about everything, including system, kernel, package managers, boot processes, Xorg, Apache, and MySQL. These log files are a useful troubleshooting tool when any security issue occurs. These files help in monitoring and analyzing security threats and vulnerabilities as well as remediate them as soon as possible. They also help in tracking the communication between one system with another system and networks.

Linux OS captures a wide range of information using multiple log files. Most logs are located in the **/var/log** directory and subdirectory in plain ASCII text format. These are system and service log files that provide information about OS-specific issues or service-specific issues. Many of them are produced by the system log daemon (syslogd) on behalf of the system and application whereas some applications produce logs directly into **/var/log** directory. To change the directory, the **cd** command is used. However, only the root user can view or access Linux log files.

Different Linux Log Files















 /var/log/messages  General message and system-related information	 /var/log/httpd/  Apache access and error logs directory
 /var/log/auth.log  Authentication logs	 /var/log/lighttpd/  Lighttpd access and error logs directory
 /var/log/kern.log  Kernel logs	 /var/log/boot.log  System boot log
 /var/log/cron.log  Crond logs (cron job)	 /var/log/mysql.log  MySQL database server log file
 /var/log/maillog  Mail server logs	 /var/log/secure or /var/log/auth.log  Authentication log
 /var/log/qmail/  Qmail log directory (more files inside this directory)	 /var/log/utmp or /var/log/wtmp  Login records file
 /var/log/yum.log  Yum command log file	

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Different Linux Log Files (Cont'd)



 /var/log/dmesg  Displays messages in this file.	 /var/log/daemon.log/  Track of running background services
 /var/log/lastlog/  holds every user's last login	 /var/log/btmp  keeps a note of all failed login attempts
 /var/log/xferlog  Records FTP file transfer sessions	 /var/log/faillog  Track info on failed logins

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Different Linux Log Files

Linux OS generates four different categories of log files: application logs, event logs, service logs, and system logs. These log files should be monitored to predict upcoming issues before they actually occur. However, it can get cumbersome to monitor and analyze all log files or to determine which file contains the required information. Therefore, to make it the process a little simpler, a few critical Linux log files are introduced here that should be monitored effectively to gather all essential information.

- **/var/log/messages** or **/var/log/syslog**: This log file contains general messages and system-related information. It stores all informational and noncritical messages across the global system such as system error messages, system startups, and shutdowns, change in the network configuration, etc. It can also log several things such as mail, cron, daemon, kern, auth, etc. This is the first place to look if things go wrong in the network/OS. For example, if there is any issue with the sound card, then you have to check the messages logged in this file. This file stores data in plain-text format that can be checked by any tool that can examine text files.
- **/var/log/auth.log** or **/var/log/secure**: This log file contains authentication logs, including both successful and unsuccessful user login attempts as well as authentication techniques. This file is beneficial if you want to examine brute-force attacks and other vulnerabilities related to user authorization mechanism.
- **/var/log/kern.log**: This file stores information that is logged by the kernel. It is helpful in solving kernel-related errors and warnings as well as hardware and connectivity problems. It is also useful in troubleshooting a custom-built kernel.
- **/var/log/cron.log**: This file contains information about all Crond-related messages (cron jobs). For example, when the cron daemon begins the cron job, all related information about successful or failed execution is logged on to this file. This file is helpful for solving issues with scheduled cron.
- **/var/log/maillog** or **/var/log/mail.log**: This file stores information related to mail servers. This file is useful when checking information regarding postfix, smtpd, MailScanner, and other email-related services. It keeps records of all emails that are sent or received within a time zone. In addition, it helps to examine failed delivery problems and detecting spamming attempts blocked by the mail server.
- **/var/log/qmail/**: It is a directory that stores information related to qmail logs. This directory is helpful when trying to track all emails sent through a qmail system, if the list of every message transmitted by the server is needed, or the number of messages processed needs to be determined.
- **/var/log/httpd/**: It is a directory that stores information related to the Apache web server. Apache web server stores information in two log files: `access_log` and `error_log`. This directory provides detailed information about events and errors raised during processing httpd requests. It keeps records of every page or file that is provided or loaded by Apache and also stores the IP address and user ID of every client that made a connection to the server. It also logs the status of access requests and whether a response was given or not.
- **/var/log/lighttpd/**: It is a directory that stores information related to light HTTPD `access_log` and `error_log`.
- **/var/log/boot.log**: This file stores all information related to system booting. The booting messages are sent by system initialization script, `/etc/init.d/bootmisc.sh`, to this log file. This file is helpful when trying to troubleshoot problems related to

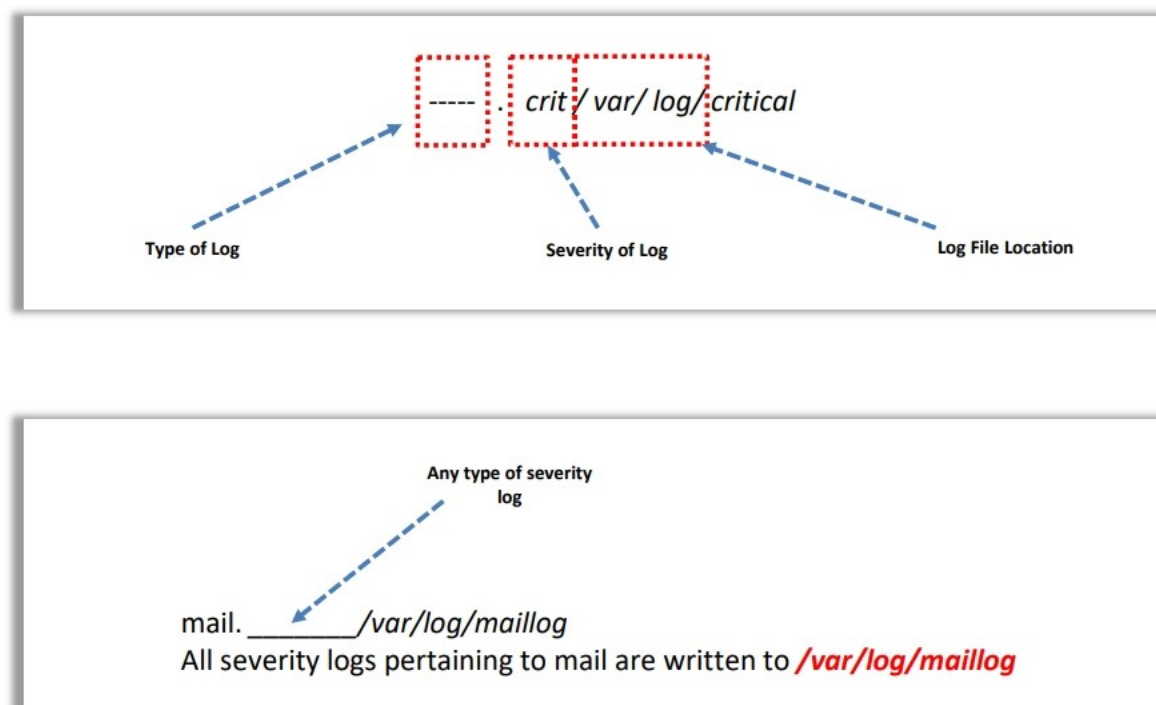
improper shutdowns, booting failures, or unplanned reboots. By checking this file, the time span of system downtime that occurred due to an unexpected shutdown can be determined.

- **/var/log/mysqld.log**: This file stores all debug, failure, and success messages about [mysqld] and [mysqld_safe] daemon. It is helpful when trying to detect issues related to starting, running, and stopping of mysqld.
- **/var/log/utmp** or **/var/log/wtmp**: This file stores information related to user login/logout, and it is helpful when trying to determine the current login state.
- **/var/log/yum.log**: All information related to installation of a package using `yum` command is stored in this file, which proves useful when trying to check whether a package is installed correctly or not; it also helps in identifying and solving software installation issues.
- **/var/log/dmesg**: This log file contains Kernel ring buffer messages and information related to hardware devices and their drivers are logged here.
- **/var/log/daemon.log/**: It records the execution of background services but does not display them in graphical form.
- **/var/log/lastlog**: It contains the last logon of each user. It is a binary file that can be read using the `lastlog` command.
- **/var/log/btmp**: It maintains a record of all unsuccessful logon attempts.
- **/var/log/xferlog**: It maintains a record of FTP file transfers that contains information such as file names and FTP transfers initiated by users.
- **/var/log/faillog**: It records info on failed logins. Hence, handy for examining potential security breaches like login credential hacks and brute-force attacks.

Linux Log Format



Format of Linux log files:



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Linux Log Format

The system log file provides information about where messages are logged. It is in the following format:

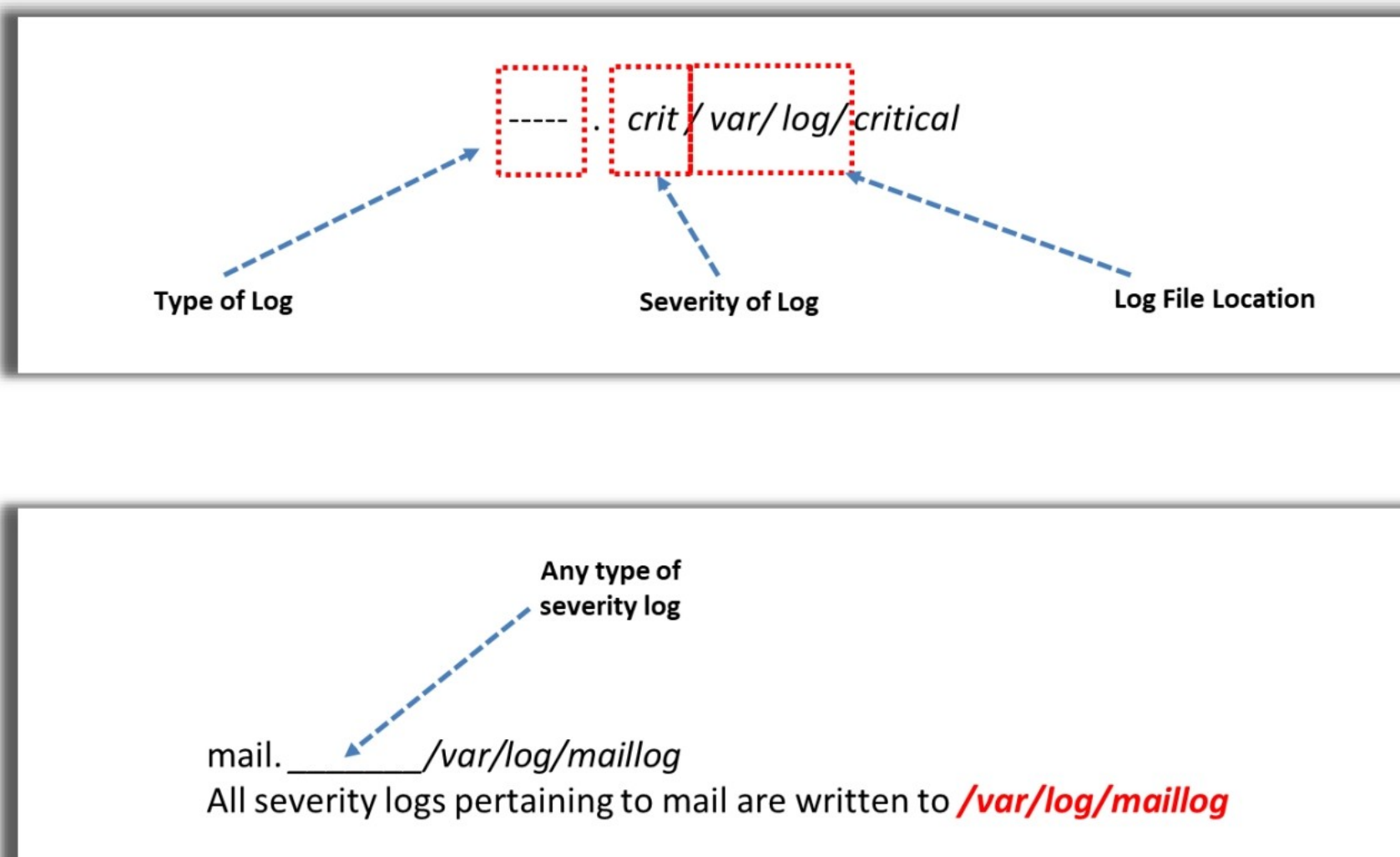


Figure 15.11: Linux log format

Consider some examples below to understand the log file format.

- **Example 1:** Log all kernel messages to the console

kern.*

/dev/console

- **Example 2:** Log anything (except mail) of level info or higher. Do not log private authentication messages

***.info;mail.none;news.none;authpriv.none;cron.none
/var/log/messages**

- **Example 3:** The authpriv file has restricted access

authpriv.*

/var/log/secure

- **Example 4:** Log all the mail messages in one place

mail.*

/var/log/maillog

Each line of the log file is divided into two portions: message selector and action field.

- Message selector represents the type of message to log. It is a combination of log type and severity level. In the above example, **kern.***, ***.info;mail.none;news.none;authpriv.none;cron.none**, **authpriv.***, and **mail.*** are the various selectors. Here, * indicates "all" such as **kern.*** all messages generated by the kernel.
- An action field describes the type of action to be applied to the message. It indicates a log file location. In the above examples, **;/dev/console**, **/var/log/messages**, **/var/log/secure**, **/var/log/maillog** represent the actions.

Severity Level and Value of Linux Logs



Severity Level	Severity Value	syslog.conf Extension	Meaning
Emergency	0	.emer	System is unusable
Alert	1	.alert	Action must be taken immediately
Critical	2	.crit	Critical conditions
Error	3	.err	Error conditions
Warning	4	.warning	Warning conditions
Notice	5	.notice	Normal but significant condition
Info	6	.info	Informational messages
Debug	7	.debug	Debug-level messages


Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Severity Level and Value of Linux Logs

The combination of Linux log file(s) and severity levels facilitates determination of what is logged and where that information is stored. When a system logger receives messages from multiple programs, it will make decisions as to what to keep and what to discard on the basis of severity levels defined by the selector. There are eight severity levels for sending a message in Linux, starting from level 0 to level 7. The highest severe message is at level 0, and the lowest severe message is at level 7.

- **Level 0—Emergency:** This level represents emergency conditions where the system comes unusable; for example, imminent system crash.
- **Level 1—Alert:** This level represents those conditions that require immediate actions; for example, a corrupted system database.
- **Level 2—Critical:** This level represents critical conditions such as a hardware error.
- **Level 3—Error:** This level represents error messages.
- **Level 4—Warning:** This level represents warning messages.
- **Level 5—Notice:** This level represents those messages that are not an error but require special attention.
- **Level 6—Information:** This level represents informational messages.
- **Level 7—Debug:** This level represents those messages that are required during debugging programs.

Monitoring and Analysis of Linux Logs



Commands Used to Monitor and Analyze Linux Log Files :

cat command

cat command displays **file contents**

```
cat [filename]
```

less command

less command displays the contents of a text file **one page (one screen) per time**

```
less [filename]
```

tail command

tail command displays **last 10 lines** from a given text file by default

```
tail [n] [filename]
```

more command

more command displays the number of lines from a text file **as much as the screen can fit**

```
more [filename]
```

head command

head command displays **first 10 lines** from a given text file by default

```
head [-n] [filename]
```

grep command

grep command is used for **searching** a specific string in a file

```
grep "search_string" [filename]
```

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Linux Logs

Monitoring and analysis of Linux logs helps determine security issues before they can significantly harm the system. Various types of commands are provided by Linux to monitor and analyze log files. Some of them are described below.

- **cat command:** cat stands for concatenate. It is one of the most important commands used in Linux OS. It reads data from the file and displays its content. It can combine the contents of two files by appending the content of the second file to the end of the first file. It can also copy the content of one file to another file. Its syntax is as follows:

cat [option] [filename]

Described below are the different types of **cat** commands.

- **cat [filename] :** This command will display the content of a given filename.
- **cat [filename1] [filename2]:** This command will display the content of filename1 and filename2.
- **cat > newfilename:** This command will create a new file with the name "newfilename."
- **cat -n [filename]:** This command displays the content of a given file with line number.
- **cat [filename1] > [filename2]:** This command copies the content of filename1 to filename2.
- **cat -s [filename]:** This command suppresses repeated empty lines.

- **cat [filename1]>>[filename2]:** This command appends the content of filename1 to the end of filename2.
- **tac [filename]:** This command displays the file in reverse order.
- **cat -E [filename]:** This command highlights the end of the line.
- **tail command:** This command displays last 10 lines from a given text file by default. It also allows options *n* number of lines and *c* number of characters. Its syntax is as follows:

tail [options] [filename(s)]

Described below are the different types of **tail** commands.

- **tail [filename]:** This command displays the last 10 lines from a given file.
- **tail [filename1] [filename2]:** This command displays the last 10 lines of both the files.
- **tail [-n] [filename]:** This command displays last *n* number of lines from a given file. For example, if 5 is used in place *n*, then only the last five lines will be displayed from a given file.
- **tail [-c] [n][filename]:** This command displays last *n* number of characters from a given file.
- **head command:** This command displays the first 10 lines from a given text file by default. It also allows options *n* number of lines and *c* number of characters. Its syntax is as follows:

head [options] [filename(s)]

Described below are the different types of **head** commands.

- **head [filename]:** This command displays the first 10 lines from a given file.
- **head [filename1] [filename2]:** This command displays the first 10 lines of both the files.
- **head [-n] [filename]:** This command displays the first *n* number of lines from a given file. For example, if 5 is used in place *n*, then only the first five lines will be displayed from a given file.
- **head [-c] [n][filename]:** This command displays the first *n* number of characters from a given file.
- **less command:** This command displays the contents of a text file, one page (one screen) per time. In case of a large size file, it will not access the complete file; instead, it will access page by page. For example, when using any text editor for reading a large size file, it will get loaded completely to main memory. However, by using **less** command, it will not load complete file; instead, it loads part by part, thus making it faster. Its syntax is as follows:

less filename

- **more command:** This command displays a number of lines from a text file—as much as the screen can fit. It helps view files in a scrollable manner and search the text, strings, and regular expressions. Its syntax is as follows:

more filename

- **grep command:** This command is used for searching a specific string in a file.

grep "search_string" [filename]

The following available options can be used to search the string:

- **-c:** It displays a count of number of lines that match a pattern.
- **-h:** It displays the matched lines but not the filenames.
- **-i:** It ignores the case for matching.
- **-l:** It displays file names' list.
- **-n:** It displays the line numbers as well as the matched line.
- **-v:** It displays all the lines without a matching pattern.
- **-w:** It matches the whole word.




LO#04: Discuss log monitoring and analysis on Mac systems

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

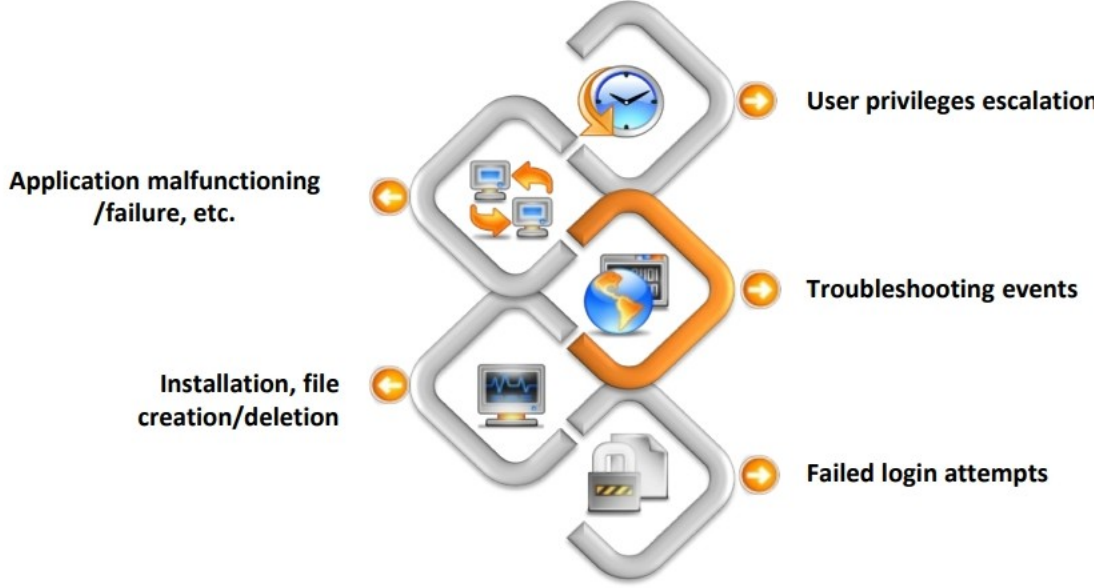
LO#04: Log Monitoring and Analysis on Mac Systems

The objective of this section is to explain monitoring and analysis of logs in Mac systems. It describes the various Mac logs, their types, log files and their formats, and how to monitor and analyze them.

Mac Logs



Mac OS events can be **configured** manually to log activities such as:



Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Mac Logs

Mac OS provides efficient application programming interfaces (APIs) for collecting log messages from all levels of the system. It stores the log data in a centralized location either in memory or in a data store on disk. As with other systems, Mac system logs help in diagnosis and troubleshooting security issues with installed applications and services. Mac OS is configured manually to log activities such as application malfunctioning/failure, user privileges escalation, installation, file creation/deletion, troubleshooting events, and failed login attempts. These logs are stored in the form of plain text and can be viewed in Mac Console app.

To launch Console app, go to Finder -> Applications -> Utilities -> Console. You can also launch it using Spotlight search. Press Command + Space and then type "Console" and press Enter. The Console app will appear, which is similar to Windows Event Viewer.

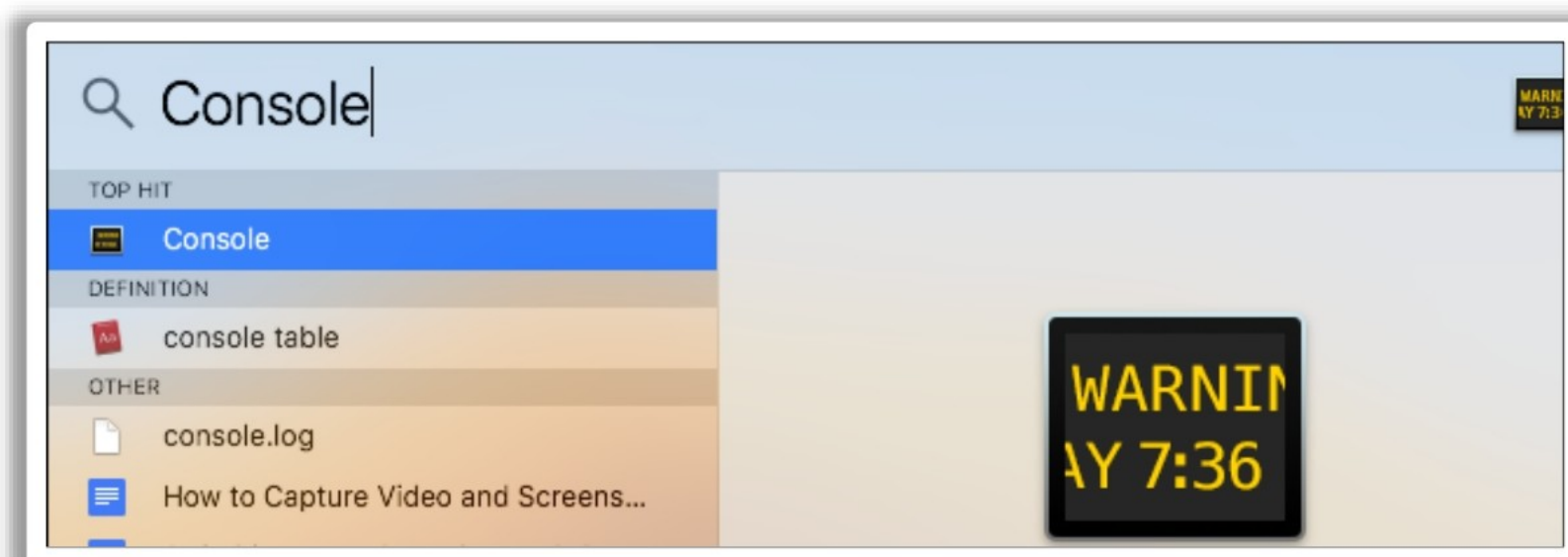


Figure 15.12: Screenshot of Console App

In the current Mac window, a list of all Console messages is showed by default. To check error messages, click on "Errors and Faults" tab in the toolbar. A particular error message can also be searched by using the search box.

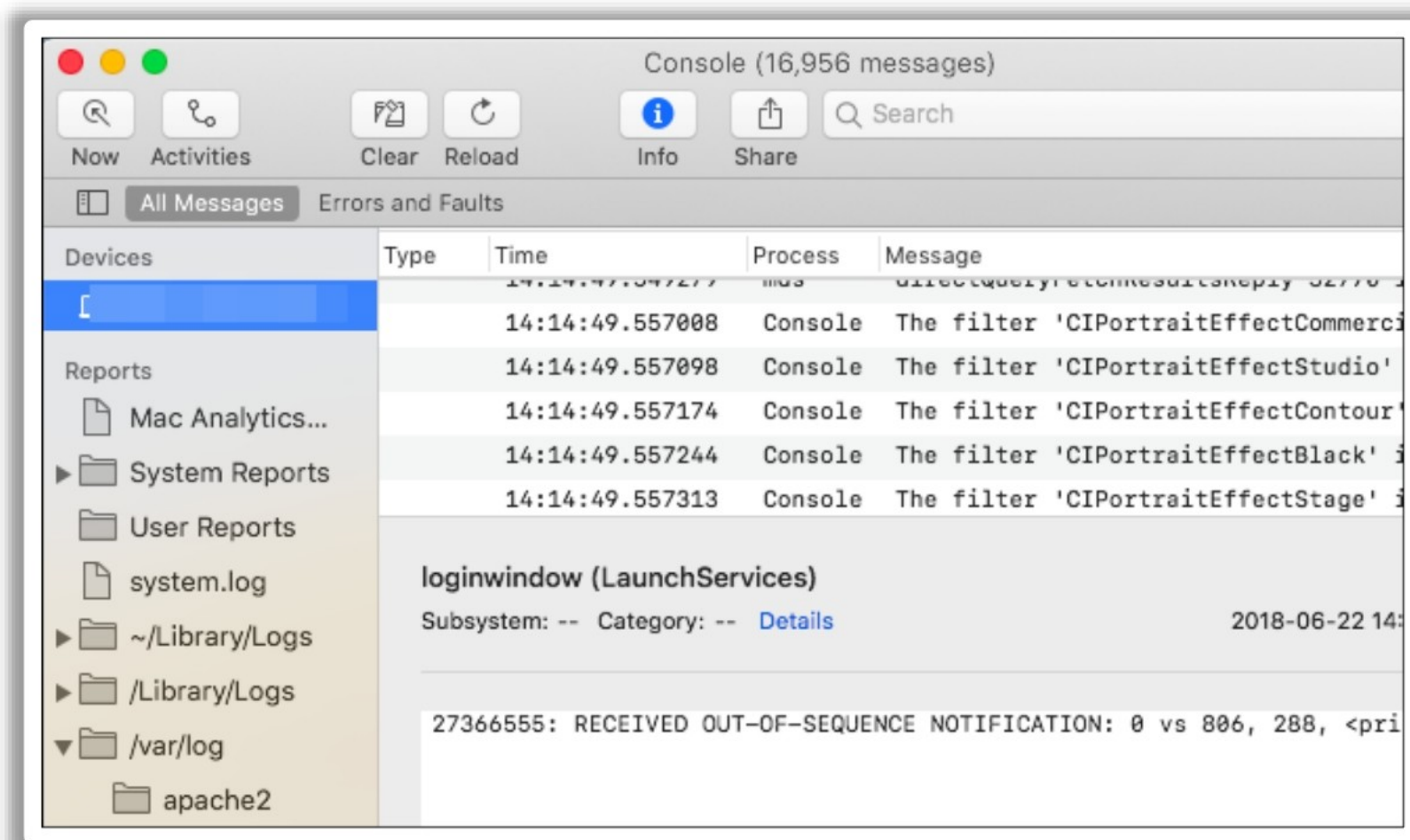
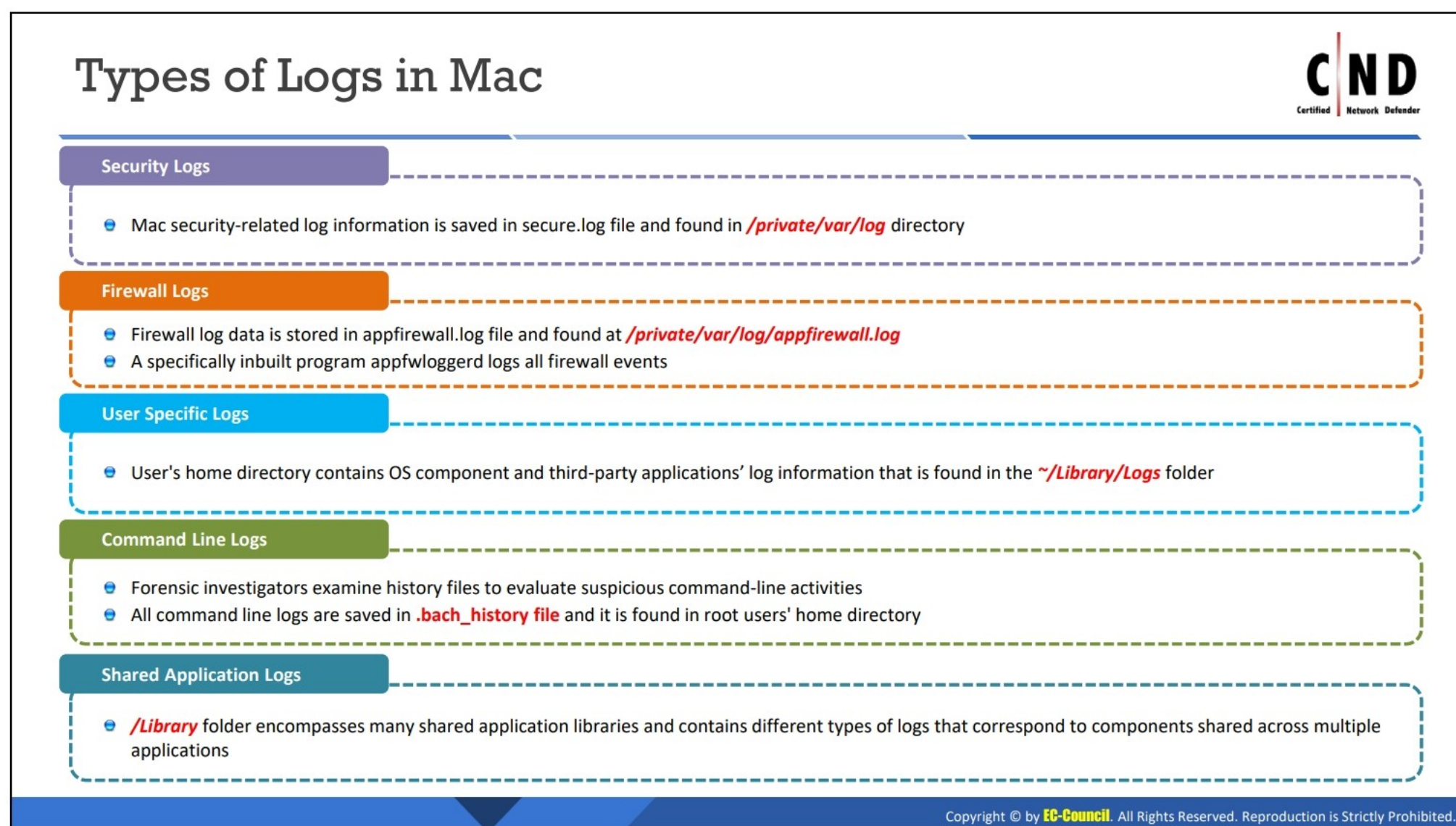


Figure 15.13: Screenshot of Mac Window



Types of Logs in Mac

Mac OS stores a variety of log files. While some of them are very detailed, only a small portion of that is used forensically; some others are harmless with direct or indirect evidence to a user's activities. Some log files represent malicious activities performed by the user, while other log files act such as indirect circumstantial evidence for identifying an attack. These types of logs are found in `/Applications/Utilities` folder in Console application.

The following are the different types of logs in Mac.

- **Security logs:** Mac security-related log information is saved in `secure.log` file, which is found in `/private/var/log` directory. It contains information about login/logout activities and helps in determining attempted and successful unauthorized activities.
- **Firewall logs:** Firewall log data is stored in `appfirewall.log` file, which is found at `/private/var/log/appfirewall.log`. `appfirewall.log` is a built-in firewall that can log a large amount of data using a program called `appfwloggerd`. The logs that are not acceptable by the application firewall are logged by `appfwloggerd`. `appfirewall.log` enables management of incoming and outgoing network traffic and helps check against abnormal/repeated attempts on ports.

- **User-specific logs:** User's home directory contains OS component and third-party applications' log information, which is found in the `~/Library/Logs` folder. These logs can be accessed only by the specific user as they are stored under that user's folder. While this guards against privacy issues, it makes troubleshooting difficult.
- **Command line logs:** Look at the command line history for each user to check command line information. All command line logs are saved in `.bash_history` file, and it is found in the root user's home directory. The history file is distinct for each shell in which the user is performing different activities. Each history file stores only 150 commands; when new commands are added, old commands automatically expire. Using the `history` command (without any arguments) displays the history, which can also be cleared using the same command through `history -c`.
- **Shared application logs:** `/Library` folder encompasses many shared application libraries and contains different types of logs that correspond to components shared across multiple applications; for example, CrashReporter logs, the information related to an application crash, and server and directory service logs.

Mac Log Files



Log file	Location	Description
crashreporter.log	/var/log/crashreporter.log	Application usage history and application crash information written to this file
access_log	/var/log/cups/access_log	Printer access log information
error_log	/var/log/cups/error_log	Printer connection information and its error logs found here
daily.out	/var/log/daily.out	Network interface history
log.nmbd	/var/log/samba/log.nmbd	Samba (Windows-based machine) connection information
Logs	~/Library/Logs	Home directory users and application-specific logs can find here
DiscRecording.log	~/Library/Logs/DiscRecording.log	Home users' CD & DVD media burning logs written to this file
DiskUtility.log	~/Library/Logs/DiskUtility.log	This file contains hard disk partitioning logs, CD/DVD burned media logs; SO/DMG images files mount, unmount history, and file permission repair history
iChatConnectionErrors	/Library/Logs/iChatConnectionErrors	Log history of iChat connection attempts. Data such as username, IP address, and date & time of the attempt
Sync	/Library/Logs/Sync	This log file gives information on synchronized Mac systems and mobile devices such as cell phones and iPods as well as their activities with date and time


Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Mac Log Files

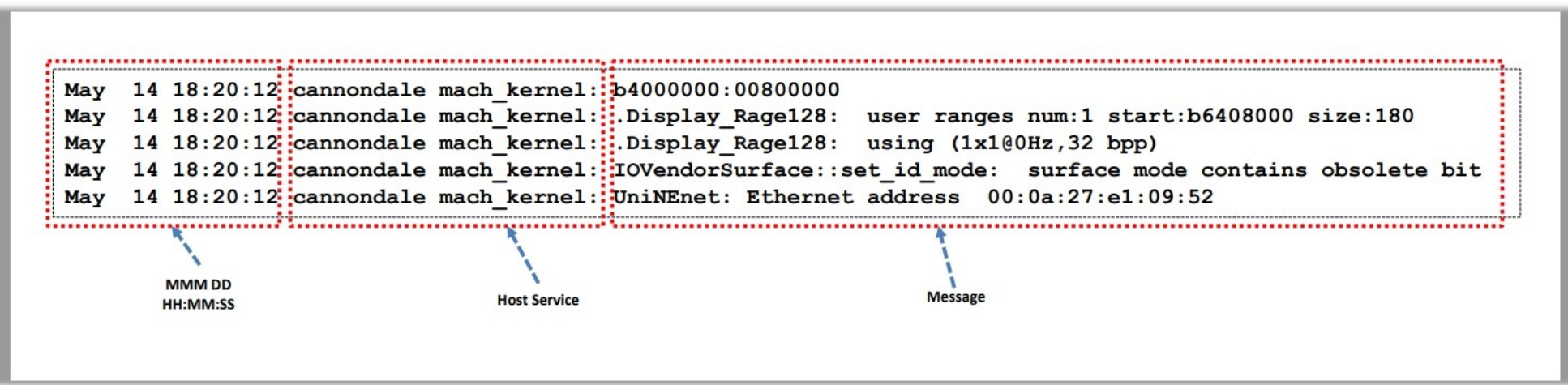
- **crashreporter.log:** Mac OS has a standard crash reporter located at **/System/Library/CoreServices/Crash Reporter.app**. It sends warning to notify that "[App] has quit unexpectedly." To check the crash report immediately, either click on "Report..." button or use Console app. To view the crash report in the Console app, click on "User Reports" in the left menu. All files related to the crash have a ".crash" extension and also include the date and crashed application in the title. The right pane contains details regarding the crash report. It contains information about what, when, and why an application/component crashed.
- **AccessLog:** It is a directive that sets the name of the access log file. In case the name of the file is not absolute, it is considered as relative to the **ServerRoot** directory. It is saved in common log format and can be utilized in producing reports on Common UNIX Printing System (CUPS) server operations. A filename can also include server name by using **%s** such as **AccessLog /var/log/cups/access_log-%s**. To send the access information to the system log, "syslog" is used in spite of a plain file such as **AccessLog syslog**. The access log file is located at **/var/log/cups/access_log** by default.

- **ErrorLog:** It is a directive that sets the name of the error log file. In case, the name of the file is not absolute, it is considered as relative to the **ServerRoot** directory. A filename can also include server name by using `%s` such as **ErrorLog /var/log/cups/error_log-%s**. To send error information to the system log, "syslog" is used in spite of a plain file such as ErrorLog syslog. The error log file is located at **/var/log/cups/error_log** by default.
- **Daily.out log:** These log files are generated based on the daily activities performed overnight when your system is running but not logged in. It stores network interface history and is located at **/var/log/daily.out**.
- **log.nmbd:** This type of file contains Samba (Windows-based machine) connection information. It is stored in **/var/log/samba/log.nmbd**.
- **Logs:** This is the home directory that contains users and application-specific logs, which are plain-text files stored in **~/Library/Logs**.
- **DiscRecording.log:** This log file stores CD and DVD media burning logs that are specific to the user of this Home directory. It is located at **~/Library/Logs/DiscRecording.log**.
- **DiskUtility.log:** This log file contains all the activities performed by the Disk Utility application such as hard disk partitioning logs, CD/DVD burned media logs, ISO/DMG images files mount, unmount history, and file permission repair history. It is located at **~/Library/Logs/DiskUtility.log**. Such a type of log cannot be rotated or cleared on a regular basis, thereby helping in detection of abnormal behavior in the system.
- **iChatConnectionErrors log:** This log file contains log history of iChat connection attempts, including data such as username, IP address, and date and time of a particular attempt. It is stored in **/Library/Logs/iChatConnectionErrors**.
- **Sync:** This log file gives information on synchronized Mac systems and mobile devices such as cell phones and iPods and their activities with date and time. It is located at **/Library/Logs/Sync**.

Log Format in Mac System



- Mac computer system follows standard **Unix log** format; most of the logs can be found in plaintext form
- Syntax: **MMM DD HH:MM:SS Host Service: Message**



```
May 14 18:20:12 cannondale mach_kernel: b4000000:00800000
May 14 18:20:12 cannondale mach_kernel: .Display_Rage128: user ranges num:1 start:b6408000 size:180
May 14 18:20:12 cannondale mach_kernel: .Display_Rage128: using (1x1@0Hz,32 bpp)
May 14 18:20:12 cannondale mach_kernel: IOVendorSurface::set_id_mode: surface mode contains obsolete bit
May 14 18:20:12 cannondale mach_kernel: UniNEnet: Ethernet address 00:0a:27:e1:09:52
```

MMM DD
HH:MM:SS

Host Service

Message

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Log Format in Mac System

Logs in Mac computer system follow standard UNIX log format, that is, **MMM DD HH:MM:SS Host Service: Message**.

Consider the below examples:

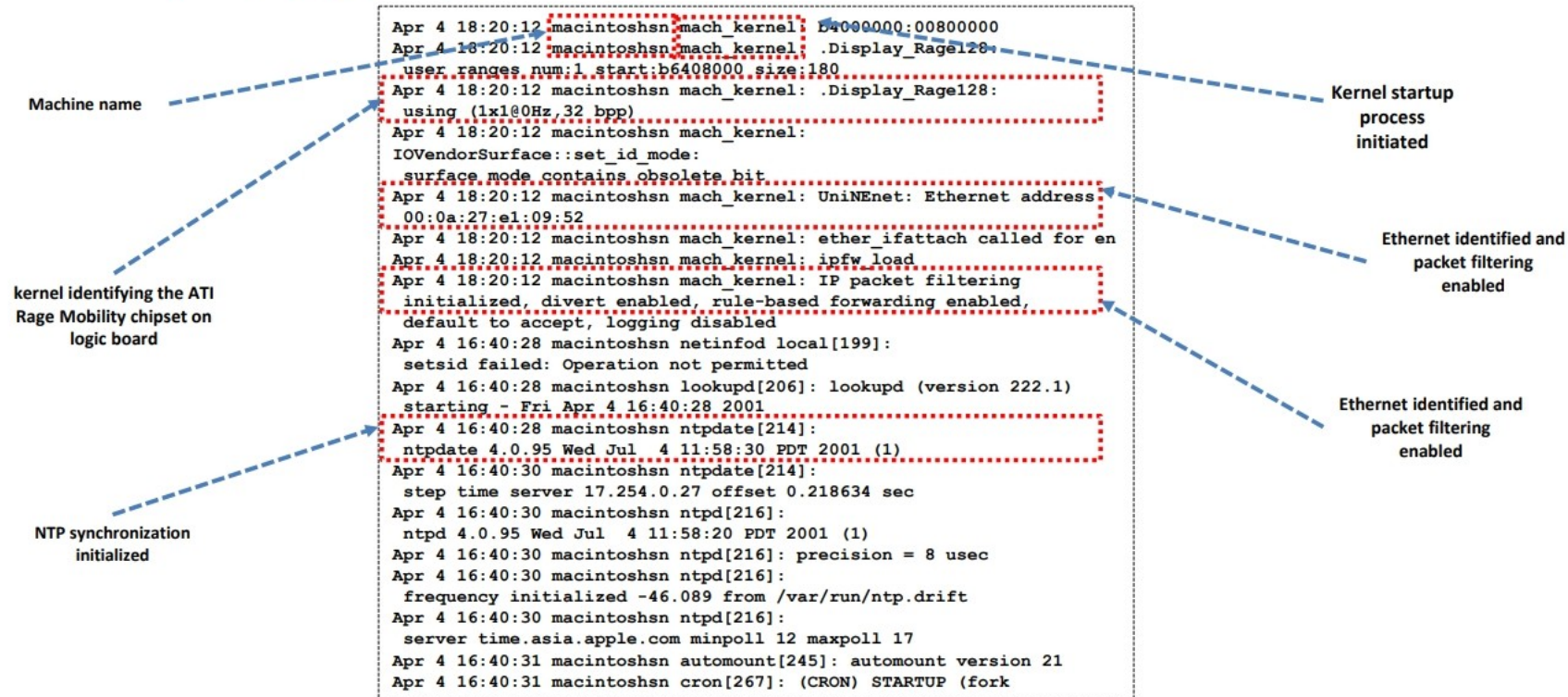
```
May 14 18:20:12 cannondale mach_kernel: b4000000:00800000
May 14 18:20:12 cannondale mach_kernel: .Display_Rage128: user
ranges num:1 start:b6408000 size:180
May 14 18:20:12 cannondale mach_kernel: .Display_Rage128: using
(1x1@0Hz,32 bpp)
May      14      18:20:12      cannondale      mach_kernel:
IOVendorSurface::set_id_mode:  surface mode contains obsolete bit
May 14 18:20:12 cannondale mach_kernel: UniNEnet: Ethernet address
00:0a:27:e1:09:52
```

In the above example, **May 14 18:20:12** represents the date and time in the **MMM DD HH:MM:SS** format; **cannondale mach_kernel** represents the host service; and **b4000000:00800000**, **.Display_Rage128: user ranges num:1 start:b6408000 size:180**, **.Display_Rage128: using (1x1@0Hz,32 bpp)**, etc. represent the messages.

Monitoring and Analysis of Mac Logs



- System Logs: **system.log** gives the details of issues regarding the whole Mac system such as DNS, networking, and Adium messages, etc.
- The file is located at **/private/var/log/system.log**



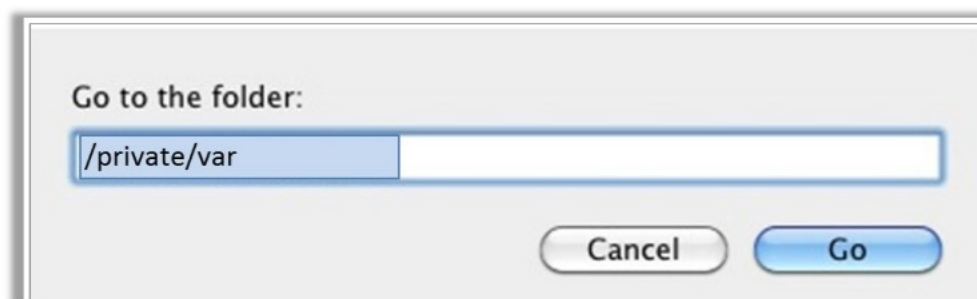
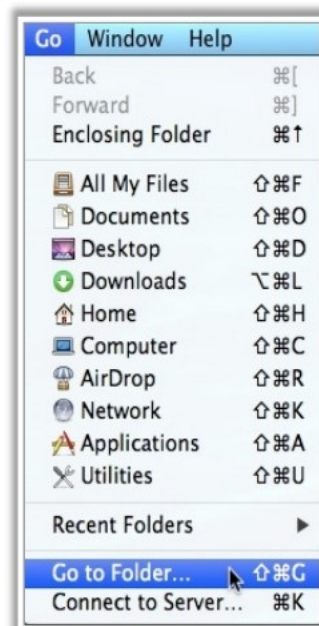
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Mac Logs (Cont'd)



Finding Logs Using "Go to folder"

- Use "Go to folder" utility for opening required log folder or use "Cmd+Shift+G" key combination to open 'Go to folder' utility
- Example:** To find **application logs** such as web server, Windows sharing components, firewall (apache, samba, ipfw), etc. specify the **"/private/var"** in the "Go to the folder" box




Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

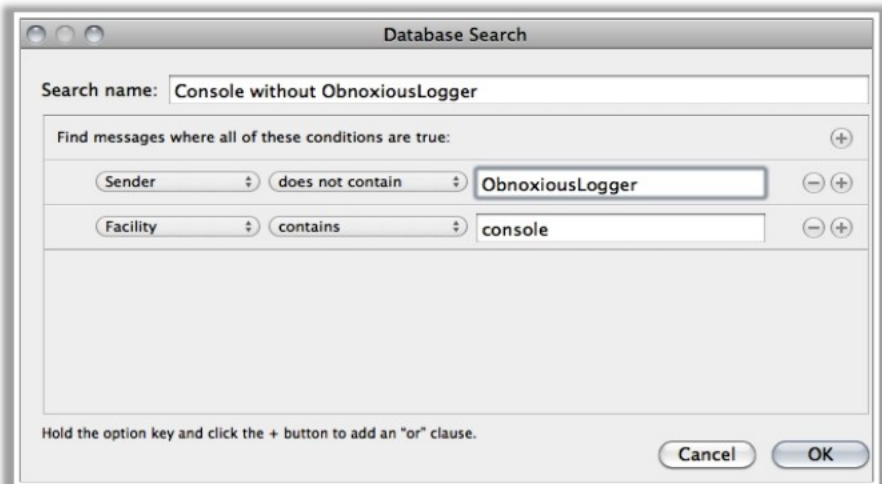
Monitoring and Analysis of Mac Logs (Cont'd)

Searching for a Particular Log

Method 1

- Search the specific interest of logs by **Edit->Find** Menu option
- Provide the additional parameters to refine the search





Method 2

- Complex log search can be done by selecting **File->New Database Search**
- Customized filter can be implement in the popped up dialog box
- Granular search is possible
 - Sender-process name
 - Facility-Sending system destination
 - Level-Severity

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Mac Logs

In Mac OS, the activities related to the system, user, and application are logged in three types of log files: system log, user log, and the application log. System log or system.log gives details on issues regarding the whole Mac system such as DNS, networking, Adium messages, etc. The file is located at `/private/var/log/system.log`. User log provides details of issues regarding user activities such as login/logout, etc. and is located in `/Users/Mac/Library/Logs`. Application logs provide details of issues regarding installed applications such as web server, Windows sharing components, and firewall and are located at `/Users/Mac/Library/Application`. The important things to notice in log files are timestamps and message.

Finding Logs Using "Go to Folder"

"Go to Folder" is the most useful Mac OS keyboard shortcut. It is used to open the required log folder. There are two ways of accessing the Go to Folder option, one is from Finder and the other is from desktop.

- Using Finder:
 - Go to the Finder in Mac operating and then click on the "Go" menu.

- Navigate down and select "Go to Folder" option.



Figure 15.14: Screenshot of Finder

- Using desktop: Use "Cmd+Shift+G" key combination from the Mac desktop to open "**Go to the folder**" utility. For example, to find the application logs such as web server, firewall (apache, samba, ipfw), etc. specify the "/private/var" in the "Go to the folder" box.

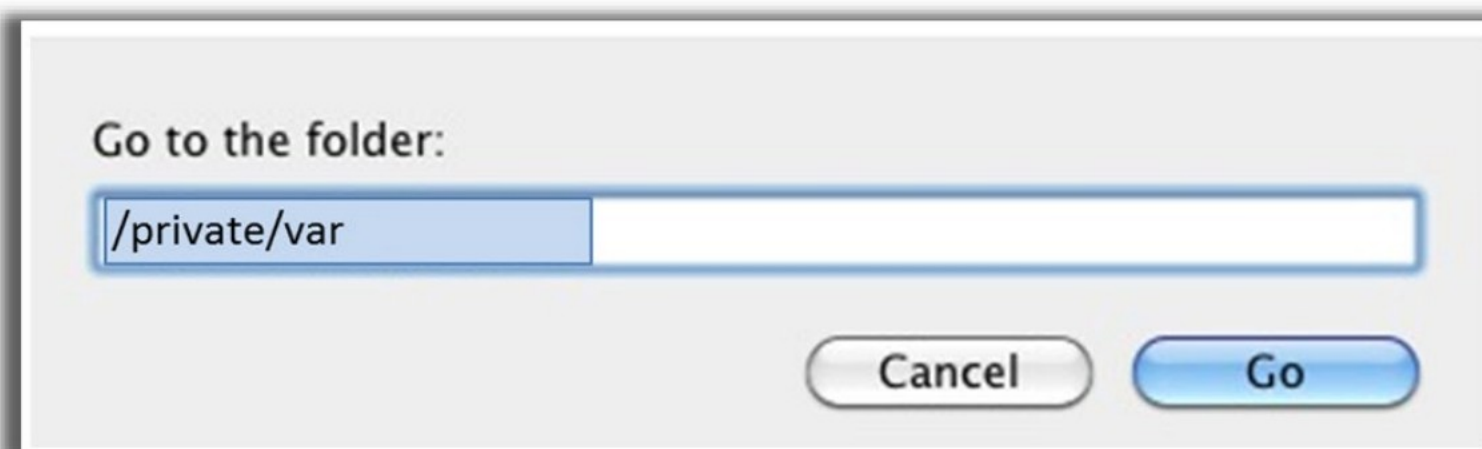


Figure 15.15: Screenshot of "Go to the folder" Dialog Box

Searching for a Particular Log

There are two methods to search a particular log in Mac OS: One is through using the Edit menu and another is from the File menu.

- **Using Edit menu:**
 - Click the Edit menu of Menu bar available at the top of the Mac desktop.
 - Navigate to the Find option and then click it.

- Provide the additional parameters to refine the search.

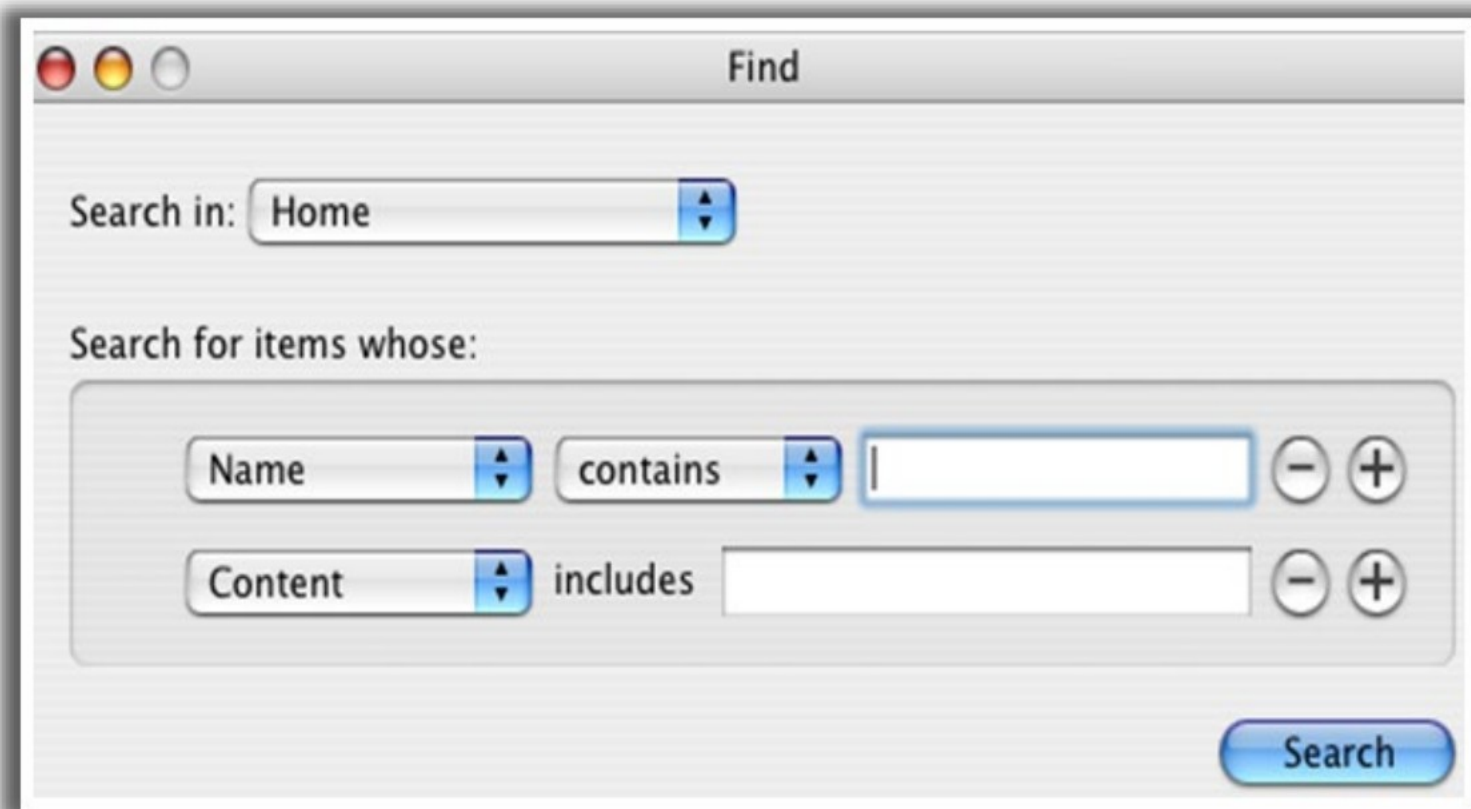


Figure 15.16: Screenshot of "Find" Dialog Box

- **Using File menu:** Complex log search can be done by using the File menu.
 - Click the File menu of Menu bar available at the top of the Mac desktop.
 - Navigate to the "New Database Search" option and then click it.
 - Implement the customized filter in the popped up dialog box.
 - Granular search is possible by giving a sender-process name, facility-sending system destination, level-severity, etc.

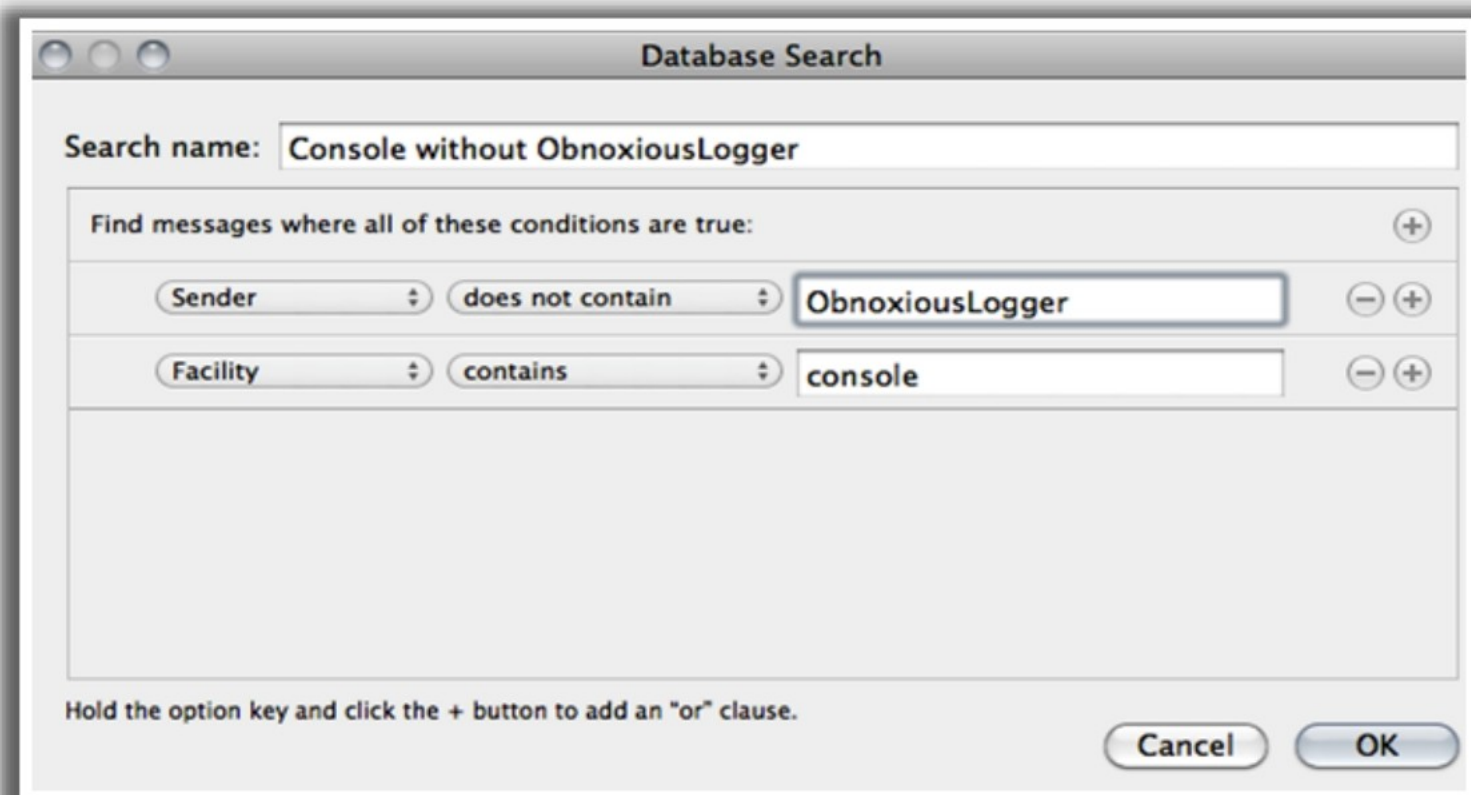


Figure 15.17: Screenshot of "Database Search" Dialog Box



LO#05: Discuss log monitoring and analysis in firewalls

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

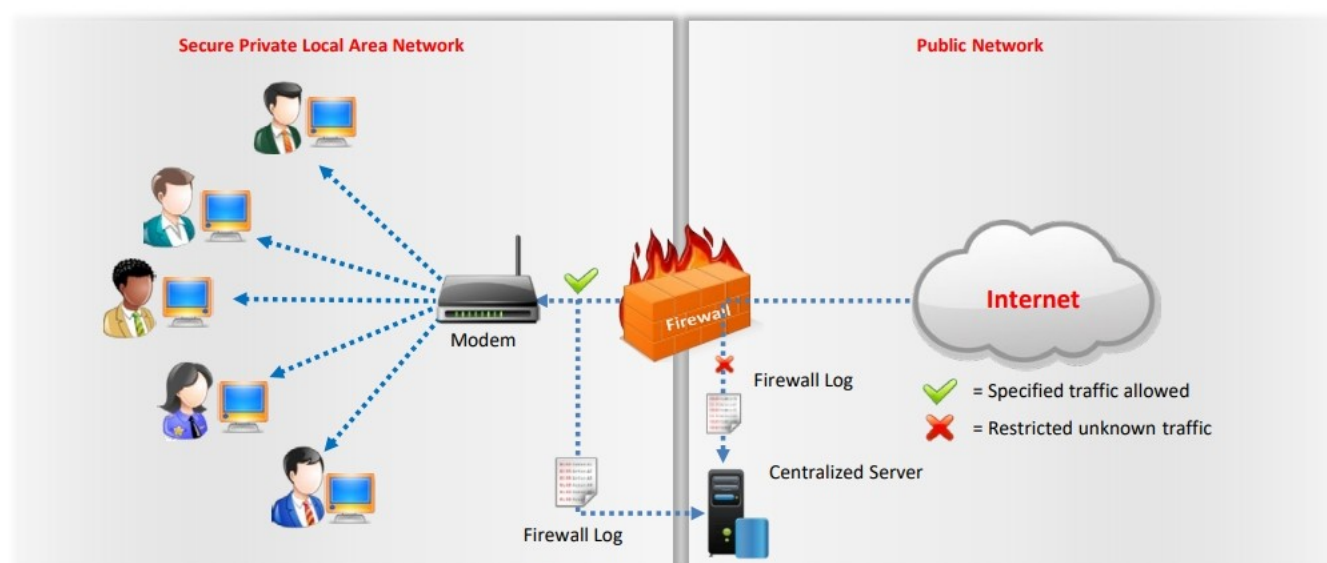
LO#05: Log Monitoring and Analysis in Firewalls

The objective of this section is to explain monitoring and analysis of firewall logs. It describes Windows, Linux, MAC, Cisco ASA, Check Point Firewall logs and demonstrates how to monitor and analyze them.

Firewall Logging



- **Logging capability** of a firewall about users' activity in a network is known as firewall logging
- Attackers leave their **footprints** when trying to pass through a firewall; investigate firewall logs to get **basic information** and to investigate the attack
- Firewall logging, regarding "allow" events, is useful for picking up on **potential security threats** to the network



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Firewall Logging

The capability of a firewall to log users' activities in a network is known as firewall logging. This logging, regarding "allow" events, is useful for capturing potential security threats to the network. Firewall logging should be enabled for security reasons, as it is the most important source for determining post attack scenarios. Some firewall logs are stored in proprietary formats, and others are polled through SNMP. Firewall logs provide various types of important information such as source and destination IP addresses, port numbers, and protocols. When the attackers trying to pass through a firewall, they leave their footprints; these should be investigated to get basic information about the attack.

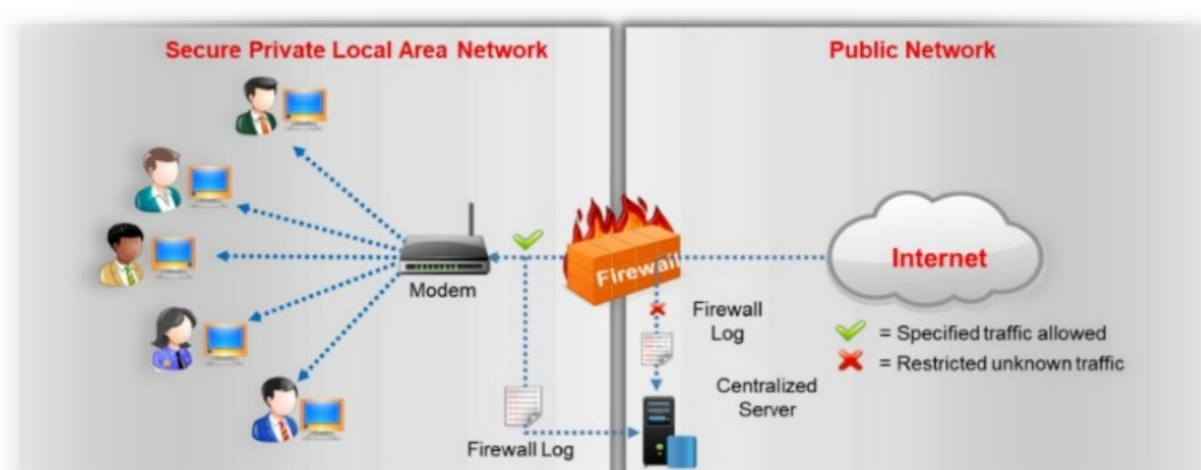


Figure 15.18: Firewall Logging

Firewall logging helps confirm whether firewall rules are working as expected. If they are not working as configured, they need to be debugged to determine the causes.

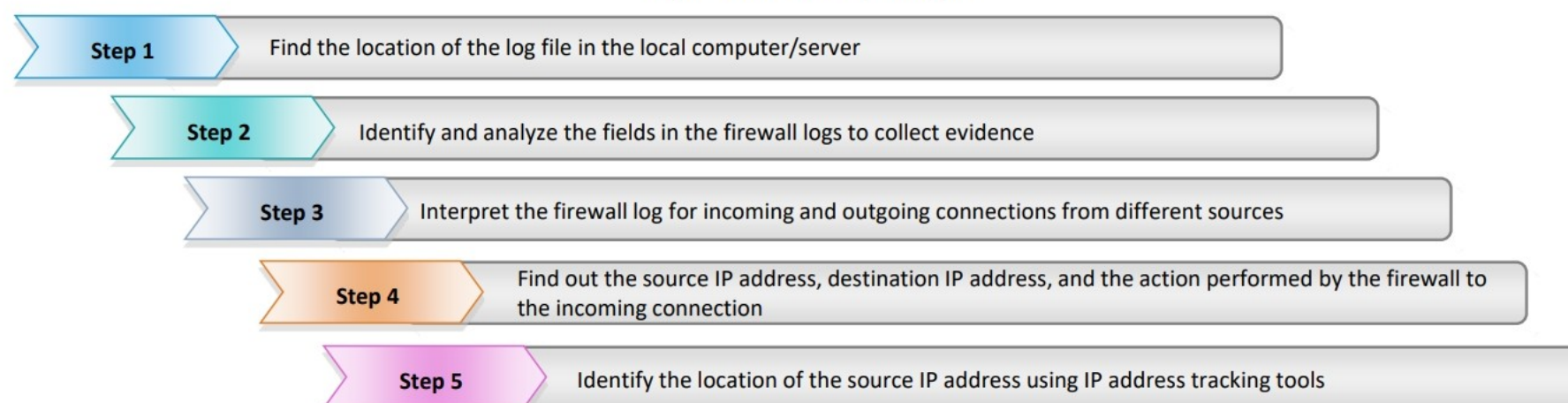
Firewall logging supports multiple levels of logging to handle the most critical events first. The levels of logging are labeled from level 0 to level 7. The events at level 0 are of the greatest importance and those at level 7 are of least importance. They are arranged in the following order: emergency, alert, critical, error, warning, notification, informational, and debugging.

Monitoring and Analysis of Firewall Logs



- Convert the firewall logs in to a **standard format** (normalization) as it simplifies the reviewing and analyzing process
- Reviewing** and **analyzing** the firewall logs lists out the **source IP addresses** that accessed the network, bandwidth used, events occurred, etc.

Steps of Firewall Log Analysis



Note : Firewall logs are recorded only when firewall logging is enabled

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Firewall Logs

Firewall logs contain a variety of important information. By monitoring and analyzing these logs, the source IP addresses that accessed the network, bandwidth used, events occurred, etc. can be known. Unauthorized connection attempts, port scan attempts, actions from compromised systems, etc. can also be detected and documented. However, the firewall logs need to be converted into a standard format (normalization), which simplifies the reviewing and analysis. The very initial step during monitoring and analysis of firewall logs is to set the proper logging levels and apply a log maintenance policy. After doing this, determine the items needed to detect abnormal or malicious activities.

IP addresses and port numbers play an important role in establishing a connection through the firewall. The IP addresses determine the various systems involved during transmission, and port numbers specify the types of applications or services that are being used. By monitoring the port numbers logged and their corresponding services, suspicious activities can be identified. The following are common items to look for in a firewall's log:

- IP addresses that are rejected and dropped
- Unsuccessful logins to the firewall and other critical servers
- Suspicious outbound activities from internal servers
- Source-routed packets
- Ports on which no application is running
- Stop/start/restart of firewall
- Change in firewall configuration

- Tear down in connection

Firewall Log Analysis

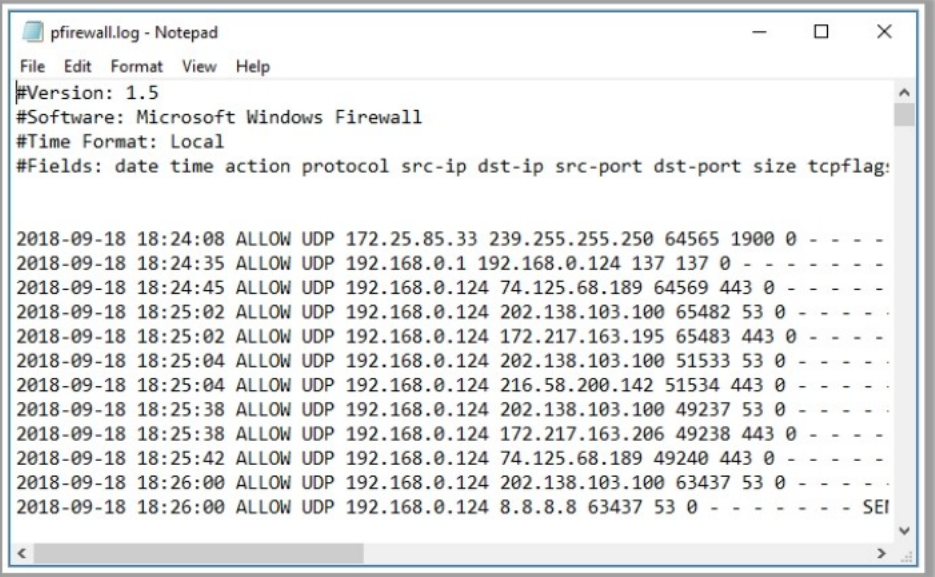
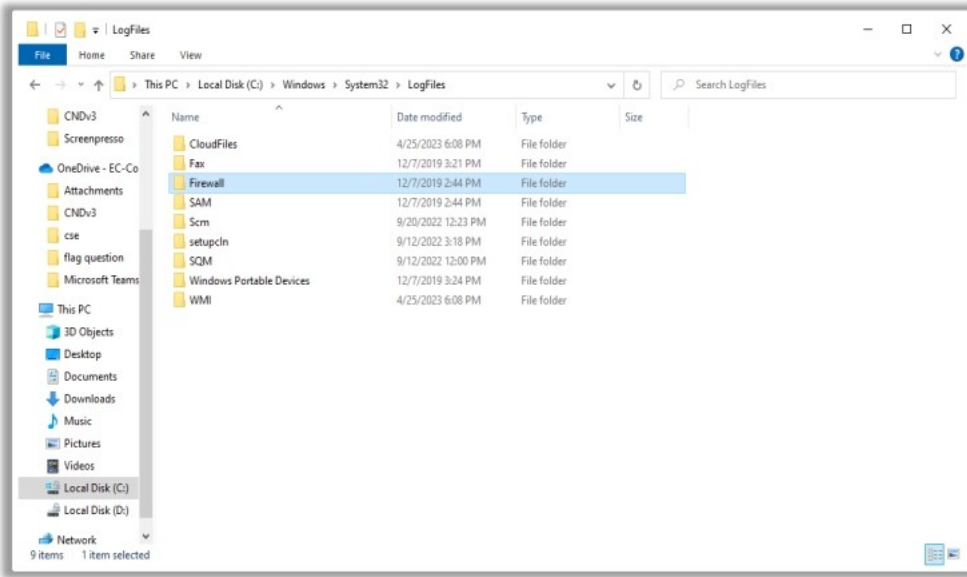
The following steps are used to analyze firewall logs:

- Find the location of the log file in the local computer/server.
- Identify and analyze the fields in the firewall logs to collect evidence.
- Interpret the firewall log for incoming and outgoing connections from different sources.
- Find out the source IP address, destination IP address, and the action performed by the firewall to the incoming connection.
- Identify the location of the source IP address using IP address tracking tools.

Windows Defender Firewall Log

- Windows Defender Firewall (if enabled) logs all activities occurred in a **network/ system**
- Whenever an attacker tries to break through Windows Defender Firewall, the details of the entry are **recorded** in a log file

- Location of Windows Defender Firewall log
 - Default firewall log location in windows is `C:\Windows\System32\LogFiles\Firewall`
 - Open the file named as **pfirewall.log**



```
File Edit Format View Help
#Version: 1.5
#Software: Microsoft Windows Firewall
#Time Format: Local
#Fields: date time action protocol src-ip dst-ip src-port dst-port size tcpflag:

2018-09-18 18:24:08 ALLOW UDP 172.25.85.33 239.255.255.250 64565 1900 0 - - -
2018-09-18 18:24:35 ALLOW UDP 192.168.0.1 192.168.0.124 137 137 0 - - -
2018-09-18 18:24:45 ALLOW UDP 192.168.0.124 74.125.68.189 64569 443 0 - - -
2018-09-18 18:25:02 ALLOW UDP 192.168.0.124 202.138.103.100 65482 53 0 - - -
2018-09-18 18:25:02 ALLOW UDP 192.168.0.124 172.217.163.195 65483 443 0 - - -
2018-09-18 18:25:04 ALLOW UDP 192.168.0.124 202.138.103.100 51533 53 0 - - -
2018-09-18 18:25:04 ALLOW UDP 192.168.0.124 216.58.200.142 51534 443 0 - - -
2018-09-18 18:25:38 ALLOW UDP 192.168.0.124 202.138.103.100 49237 53 0 - - -
2018-09-18 18:25:38 ALLOW UDP 192.168.0.124 172.217.163.206 49238 443 0 - - -
2018-09-18 18:25:42 ALLOW UDP 192.168.0.124 74.125.68.189 49240 443 0 - - -
2018-09-18 18:26:00 ALLOW UDP 192.168.0.124 202.138.103.100 63437 53 0 - - -
2018-09-18 18:26:00 ALLOW UDP 192.168.0.124 8.8.8.8 63437 53 0 - - - SEI
```

Note : Windows firewall logging should be enabled to record firewall logs

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Windows Defender Firewall Log

Windows Firewall (if enabled) logs all activities in a network/system. This log is useful in identifying suspicious and malicious activities, but it does not provide information to monitor the source of activity. It is also not beneficial when trying to determine the security status of the network. It is a plain-text file that can be viewed by any text editor such as Notepad. Every time when an attacker tries to break through Windows Firewall, the details of the entry are recorded in a log file. By default, Windows Firewall log is disabled and it does not log any of its actions. To enable Windows Firewall logs,

- Press "**Win key + R**," a Run box will open. In that box, type **wf.msc** and press Enter. The "Windows Defender Firewall with Advanced Security" window will appear.
- Click on "Properties" option located on the right pane of the window.

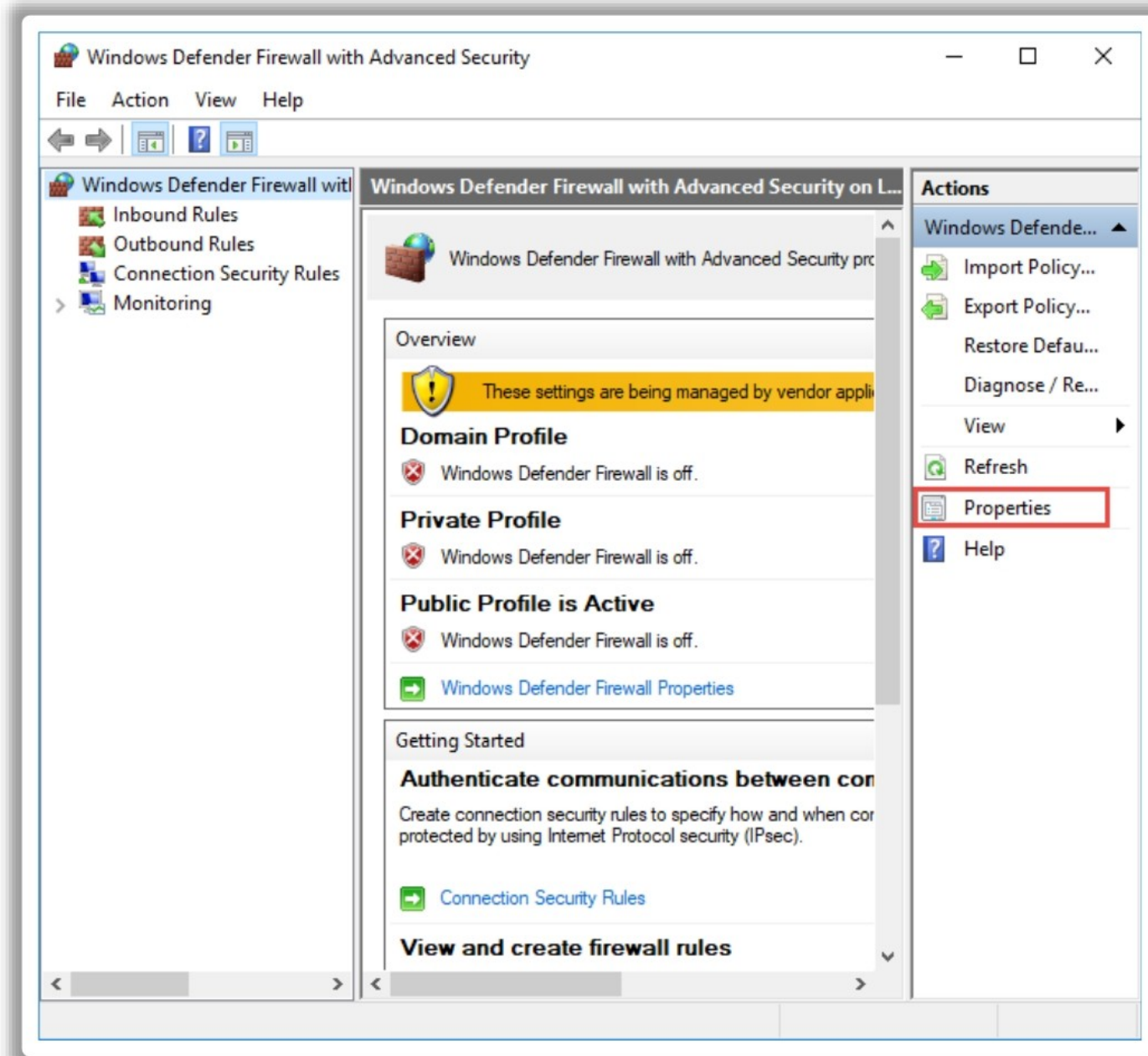


Figure 15.19: Screenshot of "Windows Defender Firewall with Advanced Security" Window

- Once a new dialog box appears, click the "Private Profile" tab and then select "Customize" available in "Logging" portion of a dialog box.

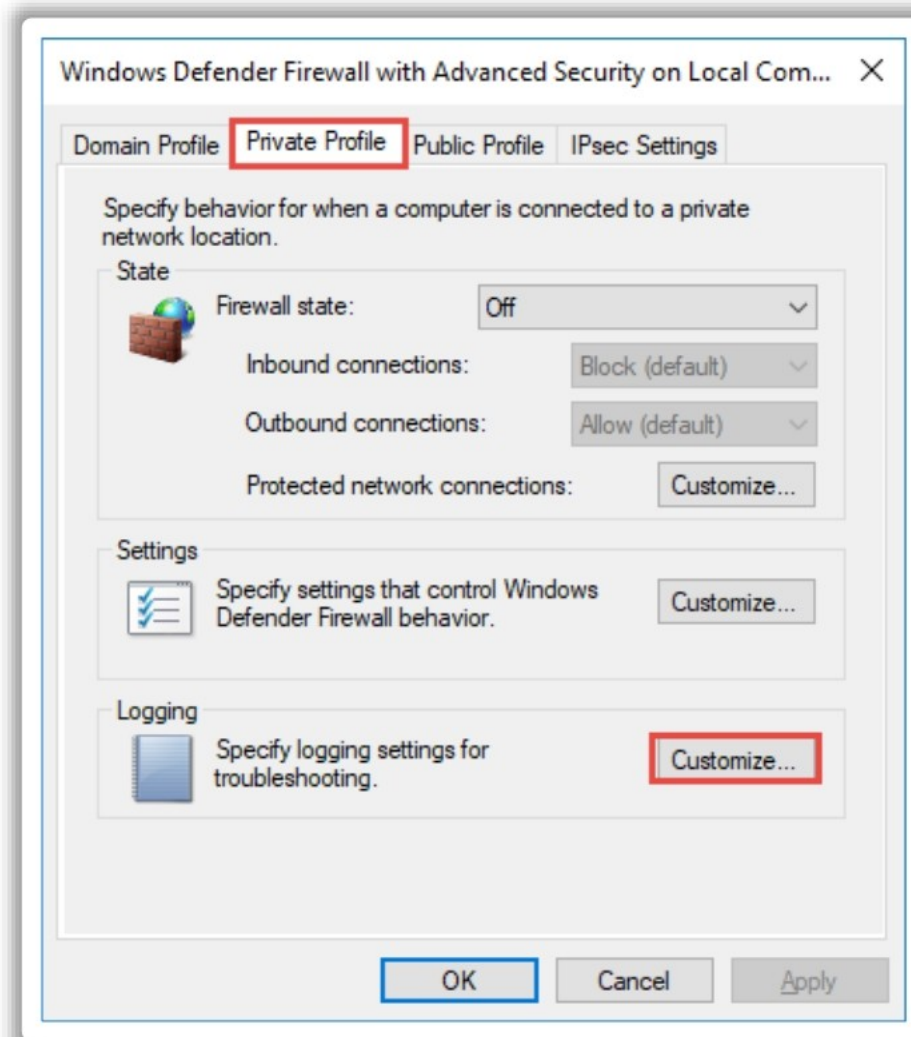


Figure 15.20: Screenshot of "Windows Defender Firewall with Advanced Security on Local Computer" Window

- A new dialog box will appear where you can set the location of log entries, maximum log size, whether to log only dropped packets, successful connection, or both. By default, the location of Windows log entries is %SystemRoot%\System32\LogFiles\Firewall\Pfirewall.log, and it stores up to 4 MB of data. If the log size limit is changed, it will affect the performance of the system. Therefore, it is suggested to enable Windows Defender Firewall logging only when you want to troubleshoot an issue actively.
- Click OK when finished.
- Now, click the "Public Profile" tab and repeat the same steps performed for "Private Profile" tab.
- On the main window "Windows Defender Firewall with Advanced Security," click the "Monitoring" option located on the left pane of the window.
- Now, click the file path next to "File Name" located under "Logging Settings" in the details pane to view the log file in Notepad.

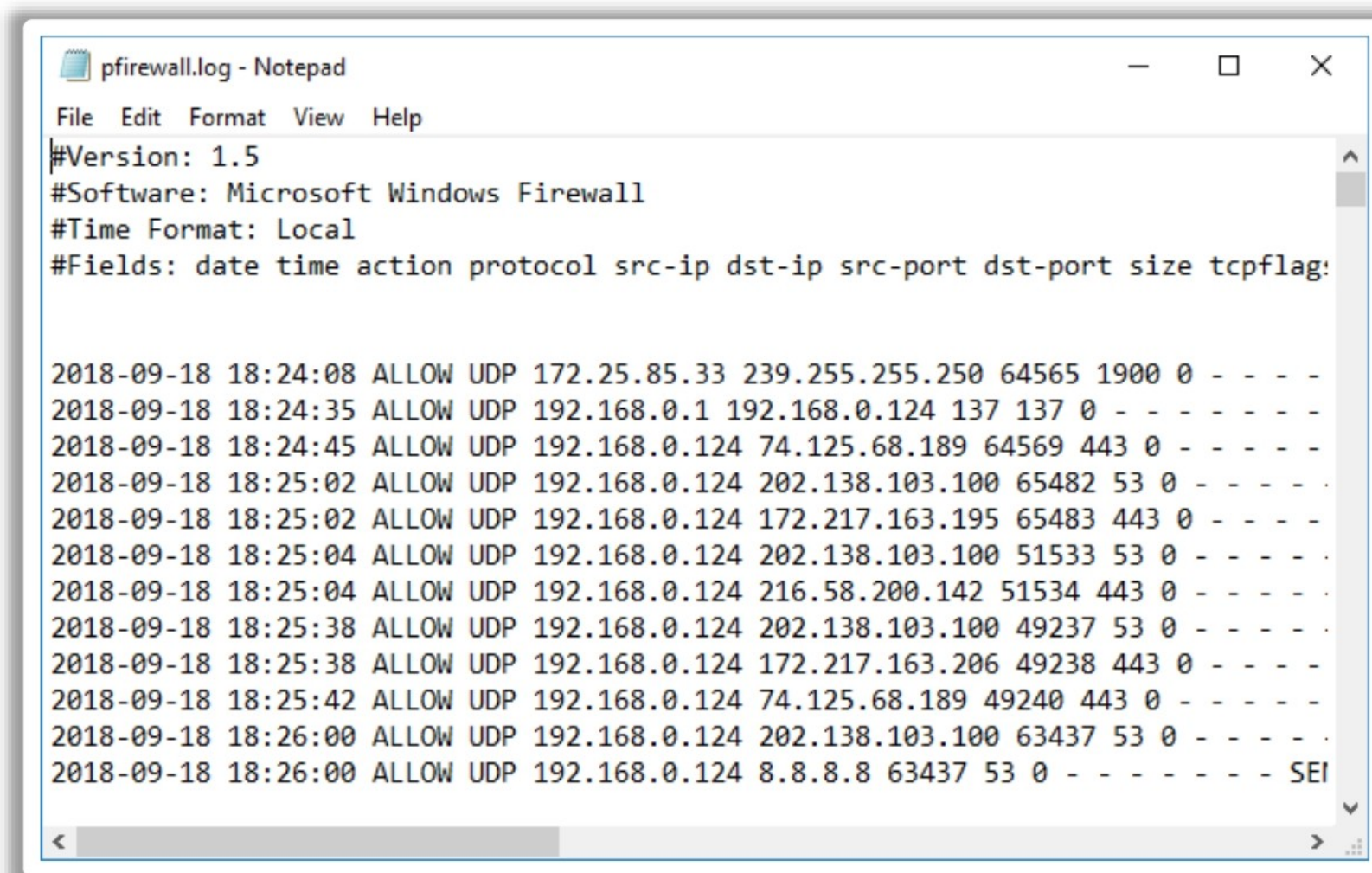


Figure 15.21: Screenshot of Log File in Notepad

Monitoring and Analysis of Windows Defender Firewall Log



Firewall Log

```
pfirewall.log - Notepad
File Edit Format View Help
#Version: 1.5
#Software: Microsoft Windows Firewall
#Time Format: Local
#Fields: date time action protocol src-ip dst-ip src-port dst-port size tcpflags tcpsyn tcpack tcpwin icmptype icmpcode info path

2018-09-18 18:24:08 ALLOW UDP 192.168.0.124 239.255.255.250 64564 1900 0 - - - - - SEND
2018-09-18 18:24:08 ALLOW UDP 172.25.85.33 239.255.255.250 64565 1900 0 - - - - - SEND
2018-09-18 18:24:35 ALLOW UDP 192.168.0.1 192.168.0.124 137 137 0 - - - - - RECEIVE
2018-09-18 18:24:45 ALLOW UDP 192.168.0.124 74.125.68.189 64569 443 0 - - - - - SEND
2018-09-18 18:25:02 ALLOW UDP 192.168.0.124 202.138.103.100 65482 53 0 - - - - - SEND
2018-09-18 18:25:02 ALLOW UDP 192.168.0.124 172.217.163.195 65483 443 0 - - - - - SEND
2018-09-18 18:25:04 ALLOW UDP 192.168.0.124 202.138.103.100 51533 53 0 - - - - - SEND
2018-09-18 18:25:04 ALLOW UDP 192.168.0.124 216.58.200.142 51534 443 0 - - - - - SEND
2018-09-18 18:25:38 ALLOW UDP 192.168.0.124 202.138.103.100 49237 53 0 - - - - - SEND
2018-09-18 18:25:38 ALLOW UDP 192.168.0.124 172.217.163.206 49238 443 0 - - - - - SEND
2018-09-18 18:25:42 ALLOW UDP 192.168.0.124 74.125.68.189 49240 443 0 - - - - - SEND
2018-09-18 18:26:00 ALLOW UDP 192.168.0.124 202.138.103.100 63437 53 0 - - - - - SEND
2018-09-18 18:26:00 ALLOW UDP 192.168.0.124 8.8.8.8 63437 53 0 - - - - - SEND
```

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Windows Defender Firewall Log

From the available firewall log information, only part of the information is important for analysis and monitoring for malicious activity or for debugging application failures. During analysis, if any suspicious activity is detected, then open the firewall log file in any text editor (Notepad, by default) to troubleshoot the issue.

Windows Defender Firewall log file is divided into two parts: header and body. The header of the log describes the static information regarding the log version and the available fields. The body of the log displays the compiled data of network traffic that is trying to move through the firewall. This list is dynamic and keeps on adding new log entries at the bottom of the log. If there is no value for a field, it is represented by (-).

```
pfirewall.log - Notepad
File Edit Format View Help
#Version: 1.5
#Software: Microsoft Windows Firewall
#Time Format: Local
#Fields: date time action protocol src-ip dst-ip src-port dst-port size tcpflags tcpsyn tcpack tcpwin icmptype icmpcode info path

2018-09-18 18:24:08 ALLOW UDP 192.168.0.124 239.255.255.250 64564 1900 0 - - - - - SEND
2018-09-18 18:24:08 ALLOW UDP 172.25.85.33 239.255.255.250 64565 1900 0 - - - - - SEND
2018-09-18 18:24:35 ALLOW UDP 192.168.0.1 192.168.0.124 137 137 0 - - - - - RECEIVE
2018-09-18 18:24:45 ALLOW UDP 192.168.0.124 74.125.68.189 64569 443 0 - - - - - SEND
2018-09-18 18:25:02 ALLOW UDP 192.168.0.124 202.138.103.100 65482 53 0 - - - - - SEND
2018-09-18 18:25:02 ALLOW UDP 192.168.0.124 172.217.163.195 65483 443 0 - - - - - SEND
2018-09-18 18:25:04 ALLOW UDP 192.168.0.124 202.138.103.100 51533 53 0 - - - - - SEND
2018-09-18 18:25:04 ALLOW UDP 192.168.0.124 216.58.200.142 51534 443 0 - - - - - SEND
2018-09-18 18:25:38 ALLOW UDP 192.168.0.124 202.138.103.100 49237 53 0 - - - - - SEND
2018-09-18 18:25:38 ALLOW UDP 192.168.0.124 172.217.163.206 49238 443 0 - - - - - SEND
2018-09-18 18:25:42 ALLOW UDP 192.168.0.124 74.125.68.189 49240 443 0 - - - - - SEND
2018-09-18 18:26:00 ALLOW UDP 192.168.0.124 202.138.103.100 63437 53 0 - - - - - SEND
2018-09-18 18:26:00 ALLOW UDP 192.168.0.124 8.8.8.8 63437 53 0 - - - - - SEND
```

Figure 15.22: Screenshot of “Windows Defender Firewall” Log File

Header

The following table explains the information present in the header of Windows Defender Firewall log:

Information	Description
#Version	Displays the version of Windows Defender Firewall security log; for example, #Version: 1.5
#Software	Displays the software name creating the log; for example, #Software: Microsoft Windows Firewall
#Time	Displays timestamps of the login local time; for example, #Time Format: Local
#Fields	Displays static list of fields that are available for security log entries (If available); for example, #Fields: date time action protocol src-ip dst-ip src-port dst-port size tcpflags tcpsyn tcpack tcpwin icmp type icmpcode info path

Table 15.5: Information in the Header Part of "Windows Defender Firewall" Log

Body

The following table explains the information present in the body part of Windows Defender Firewall log:

Fields	Description
Date	Displays the date of the log transaction in the YYYY-MM-DD format; for example, 2015-06-19
Time	Displays the time of the log transaction in the HH:MM:SS format. The hours are displayed in 24-h format; for example, 22:00:32
Action	Displays which operation was noticed by Windows Defender Firewall. Available options are OPEN (indicates the connection is opened), CLOSE (indicates the connection is closed), DROP (indicates connection is dropped), OPEN-INBOUND (indicates inbound session is opened), and INFO-EVENTS-LOST indicates (events appeared but not recorded in the log)
Protocol	Displays the protocol used for communication such as TCP, UDP, or ICMP
src-ip	Displays source IP address; for example, 192.168.2.48
dst-ip	Displays destination IP address; for example, 134.170.108.224
src-port	Displays source port number of the sending computer; for example, 56092
dst-port	Displays the port number of the destination computer; for example, 443

Size	Displays packet size (bytes)
Tcpflags	Displays TCP control flags in the TCP header of an IP packet
Tcpsyn	Displays TCP sequence number in the packet
Tcpack	Displays TCP acknowledgment number in the packet
Tcpwin	Displays TCP window size (in bytes) in the packet
Icmptype	Displays a number that represents the Type field of the ICMP message
Icmpcode	Displays a number that represents the Code field of the ICMP message
Info	Displays an entry that depends on the type of action that occurred; for example, SEND
Path	Displays the direction of the communication

Table 15.6: Information in the Body Part of “Windows Defender Firewall” Log

Mac OS X Firewall Logs



- Default location of the firewall log file in Mac is **/private/var/log/**
- Log file is saved as **appfirewall.log**, open the recent log file

Field	Description
MONTH	Month of the access attempt
DAY	Day on which access attempt made
TIME	Access attempt time
HOST	Host Name
IPFW	Firewall
CODE	IPFW code
ACTION	Firewall response to an activity ("accept" or "deny")
PROTOCOL	Protocol used in the access attempt
SOURCE	Source IP address from where access attempt made
DEST	Destination IP address to which access attempt made
IN_OUT	Access direction (coming to the firewall machine, or going out)
RESULT	OK denotes access granted, ERR! denotes a denied access
HOSTNAME	Client IP address trying to get access of a given port
SERVER_PORT	Port to which access is attempted by the user
METHOD	Protocol used by the access attempt (TCP, UDP, or ICMP)
DIRECTION	Access direction (incoming or outgoing network traffic)

Note : Firewall logging should be enabled to record firewall logs

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Mac OS X Firewall Logs

Mac OS X has a built-in firewall that helps in monitoring and analysis of the various logs associated with the system firewall. Mac OS X firewall logs display the applications and services that attempted to make a connection to the Mac system. However, Mac OS X firewall is able to record firewall logs only if it is enabled. To enable Mac OS X firewall logs:

- Select "System Preferences" option from the Apple menu. A new screen will appear:

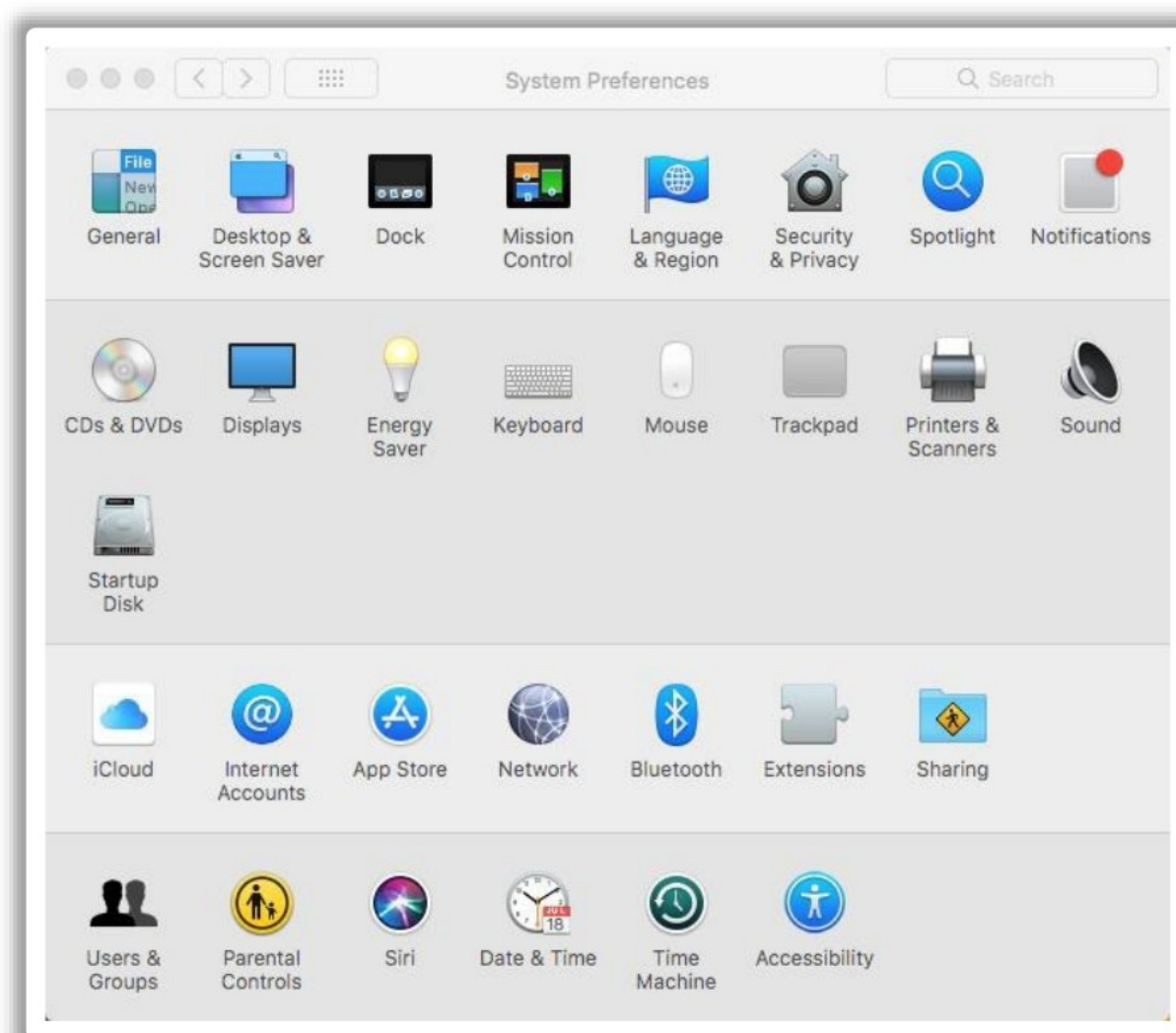


Figure 15.23: Screenshot of "System Preferences" Screen

- Click "Security & Privacy" option located under "Personal" section. A new dialog box opens.
- In the dialog box, click the "Firewall" tab and then click the lock icon and provide the administrator username and password.
- Click the "Turn On Firewall" button to turn the firewall on. When the firewall turns on, it displays a green light and the "Firewall: On" message.

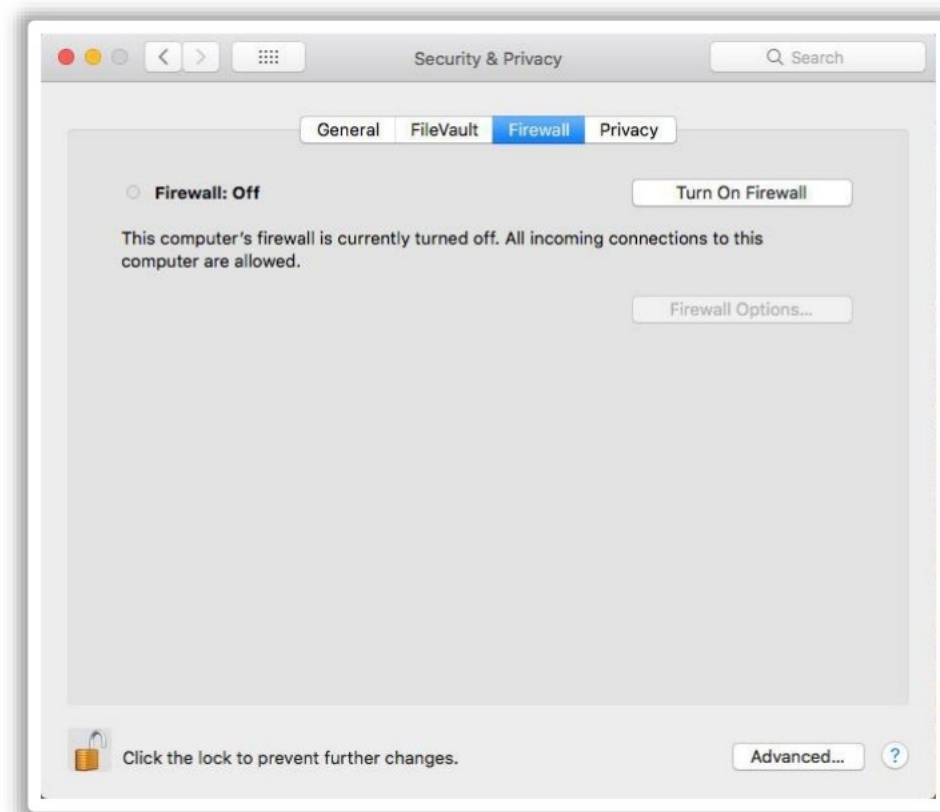


Figure 15.24: Screenshot of "Security & Privacy" Dialog Box

- Click "Advanced..." option located at the right bottom side of the "Security & Privacy" dialog box. A new screen will appear:

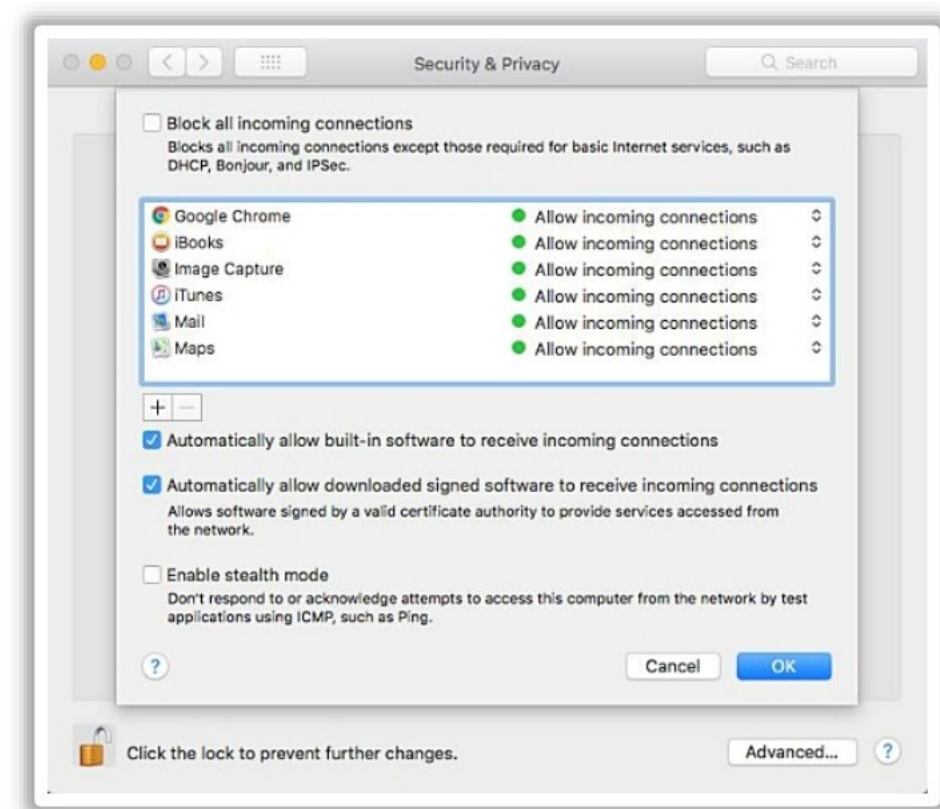



Figure 15.25: Screenshot of "Security & Privacy" Dialog Box

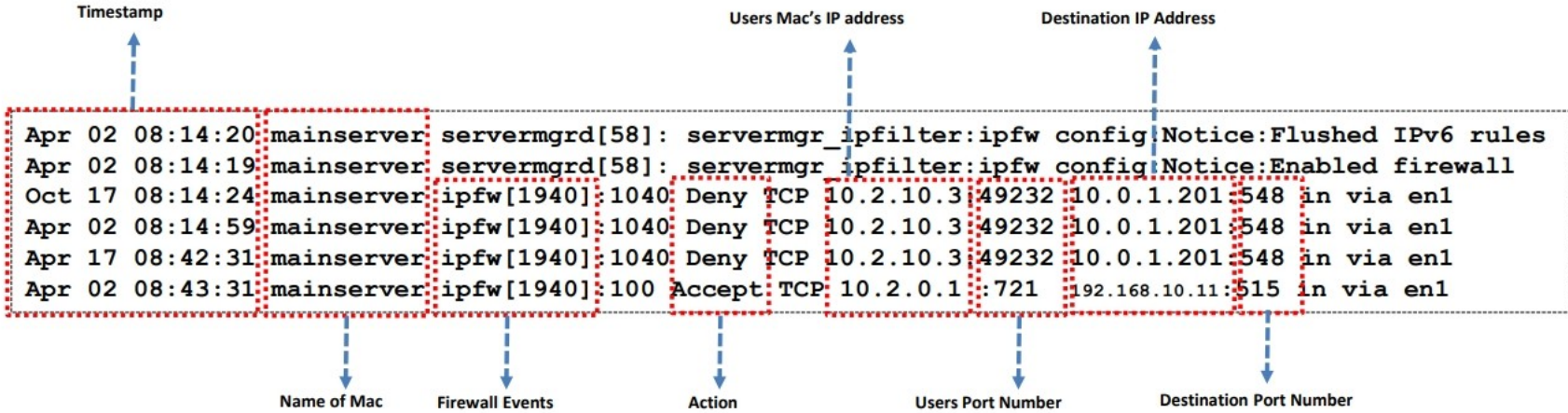
- Navigate down and check the two available options, "Automatically allow signed software to receive incoming connections" and "Enable stealth mode."
- Click OK to close "Advanced" settings dialog box.



Monitoring and Analysis of Firewall Log in Mac

■ Open the recently created log file **appfirewall.log** from the default location
 ■ Firewall log format:

<Time of entry> <Host name> <Process name and ID> <Log message> <Matching rule number> <Action> <Protocol> <Source> <Destination> <Interface>



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Firewall Logs in Mac

The following are the steps to view Mac OS X firewall logs:

- Enable the Mac firewall, if it is not enabled.
- Open Console application through Applications -> Utilities. A new screen will appear.
- Search for **/var/log** directory in the sidebar and click the disclosure triangle next to that directory.
- Click "appfirewall.log" from the sidebar to view the firewall log. The firewall log will appear into the right console panel. The log file is saved as appfirewall.log; open the most recent log file.

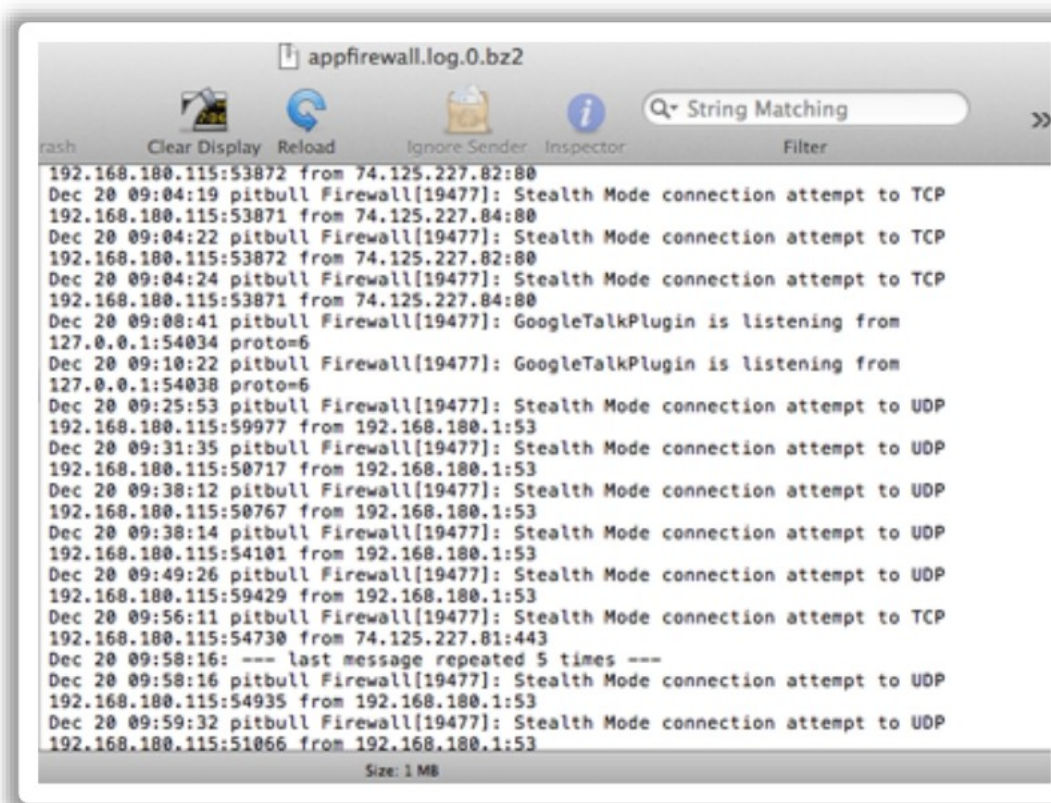


Figure 15.26: Screenshot of "appfirewall.log"

The Mac OS X firewall logs will appear in the following format:

MONTH DAY TIME HOST IPFW CODE ACTION PROTOCOL SOURCE DEST IN_OUT RESULT
HOSTNAME SERVER_PORT METHOD DIRECTION

For example:

Apr 02 08:14:20 mainserver servermgrd[58]: servermgr_ipfilter:ipfw
config:Notice:Flushed IPv6 rules

Apr 02 08:14:19 mainserver servermgrd[58]: servermgr_ipfilter:ipfw
config:Notice:Enabled firewall

Oct 17 08:14:24 mainserver ipfw[1940]:1040 Deny TCP 10.2.10.3:49232
10.0.1.201:548 in via en1

Apr 02 08:14:59 mainserver ipfw[1940]:1040 Deny TCP 10.2.10.3:49232
10.0.1.201:548 in via en1

Apr 17 08:42:31 mainserver ipfw[1940]:1040 Deny TCP 10.2.10.3:49232
10.0.1.201:548 in via en1

Apr 02 08:43:31 mainserver ipfw[1940]:100 Accept TCP 10.2.0.1 :721
192.168.10.11:515 in via en1

A Mac OS X firewall log contains the following types of fields:

Field	Description
MONTH	Month of the access attempt
DAY	Day on which access attempt made
TIME	Access attempt time
HOST	Hostname
IPFW	Firewall
CODE	IPFW code
ACTION	Firewall response to an activity ("accept" or "deny")
PROTOCOL	Protocol used in the access attempt
SOURCE	Source IP address from where access attempt made
DEST	Destination IP address to which access attempt made
IN_OUT	Access direction (coming to the firewall machine, or going out)
RESULT	OK denotes access granted, ERR! denotes a denied access
HOSTNAME	Client IP address trying to get access of a given port
SERVER_PORT	Port to which access is attempted by the user
METHOD	Protocol used by the access attempt (TCP, UDP, or ICMP)
DIRECTION	Access direction (incoming or outgoing network traffic)

Table 15.7: Types of Fields in a Mac OS X Firewall Log

Linux Firewall: iptables



- iptables is a rule-based inbuilt firewall in different versions of Linux OS
- iptables log messages to a `/var/log/messages` file through Linux syslogd daemon

Sample Firewall Log File

Timestamp of the log entry

Destination Machine IP address

Source Machine IP address

```
Sep 19 19:06:43 localhost kernel: IN=eth0 OUT= MAC=00:16:3c:03:fd:10:00:30:48:57:f2:87:08:00 SRC=69.89.31.85  
DST=206.253.165.112 LEN=60 TOS=0x00 PREC=0x00 TTL=57 ID=22657 DF PROTO=TCP SPT=55681 DPT=43 WINDOW=5840 RES=0x00  
SYN URGP=0  
Sep 19 19:06:43 localhost kernel: IN=eth0 OUT= MAC=00:16:3c:03:fd:10:00:30:48:57:f2:87:08:00 SRC=69.89.31.85  
DST=206.253.165.112 LEN=60 TOS=0x00 PREC=0x00 TTL=57 ID=59936 DF PROTO=TCP SPT=57361 DPT=5432 WINDOW=5840 RES=0x00  
SYN URGP=0  
Sep 19 19:06:43 localhost kernel: IN=eth0 OUT= MAC=00:16:3c:03:fd:10:00:30:48:57:f2:87:08:00 SRC=69.89.31.85  
DST=206.253.165.112 LEN=60 TOS=0x00 PREC=0x00 TTL=57 ID=6671 DF PROTO=TCP SPT=57972 DPT=873 WINDOW=5840 RES=0x00  
SYN URGP=0  
Sep 19 19:06:43 localhost kernel: IN=eth0 OUT= MAC=00:16:3c:03:fd:10:00:30:48:57:f2:87:08:00 SRC=69.89.31.85  
DST=206.253.165.112 LEN=60 TOS=0x00 PREC=0x00 TTL=57 ID=58039 DF PROTO=TCP SPT=57438 DPT=5432 WINDOW=5840 RES=0x00  
SYN URGP=0  
Sep 19 19:06:43 localhost kernel: IN=eth0 OUT= MAC=00:16:3c:03:fd:10:00:30:48:57:f2:87:08:00 SRC=69.89.31.85  
DST=206.253.165.112 LEN=60 TOS=0x00 PREC=0x00 TTL=57 ID=28698 DF PROTO=TCP SPT=58049 DPT=873 WINDOW=5840 RES=0x00  
SYN URGP=0  
Sep 19 19:06:49 localhost kernel: IN=eth0 OUT= MAC=ff:ff:ff:ff:ff:ff:00:16:3c:38:c5:c8:08:00 SRC=206.253.165.192  
DST=206.253.165.255 LEN=78 TOS=0x00 PREC=0x00 TTL=64 ID=0 DF PROTO=UDP SPT=137 DPT=137 LEN=58  
Sep 19 19:06:50 localhost kernel: IN=eth0 OUT= MAC=ff:ff:ff:ff:ff:ff:00:16:3c:5e:19:d0:08:00 SRC=206.253.165.168  
DST=255.255.255.255 LEN=137 TOS=0x00 PREC=0x00 TTL=128 ID=12679 PROTO=UDP SPT=17500 DPT=17500 LEN=117  
Sep 19 19:06:50 localhost kernel: IN=eth0 OUT= MAC=ff:ff:ff:ff:ff:ff:00:16:3c:5e:19:d0:08:00 SRC=206.253.165.168  
DST=206.253.165.255 LEN=137 TOS=0x00 PREC=0x00 TTL=128 ID=12680 PROTO=UDP SPT=17500 DPT=17500 LEN=117
```

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Linux Firewall: iptables

iptables is a rule-based inbuilt firewall in different versions of Linux OS. It can allow, drop, or modify network traffic coming in and going out of a system through a set of configurable table rules. It contains a set of tables that help process packets in some specific manner. These tables include multiple chains that investigate network traffic at various points. These chains have built-in or user-defined rules that describe the action to be performed on a packet.

When a packet appears, iptables matches it against the rules. If a match is found, a TARGET is given to it. A target can be another chain to match with or any of the given values.

- **Accept:** The packet is allowed to pass through.
- **Drop:** The packet is disallowed to pass through.
- **Return:** The packet is returned to the chain from where it was called in a table.

If a match is not found, the default action is followed.

iptables Log Storage

By default, iptables log entries are stored in `/var/log/messages`.

iptables Log Format and Fields

The iptables log file is displayed in the following format:

- <Date><Time HH:MM:SS><Machine Name><Action><IN><OUT><SRC-IP><DEST-IP>
<LEN><TOS><PREC><TTL><ID><FRAG><PROTO><SPT><DPT><WINDOW><RES><SYN><URGP>
- Eg: June 16 21:12:56 FW2 kernel : RULE_08a-ACCEPT IN=eth1 OUT eth0 SRC=192.42.93.30 DST= 192.168.1.102 LEN=96 TOS=0x00 PREC=0x00 TTL=64 ID=61495 DF PROTO=UDP SPT=53981 DPT=127 WINDOW=32767 RES=0x00 SYN URGP=0

iptables Log Fields

Field	Description	Field	Description
Date	Displays the date of the log transaction	PREC	TOS field's top 3 precedence bits
Time	Displays the time of the log transaction (HH:MM>SS)	TTL	Time to live
Machine Name	Name of the machine	ID	Packet's datagram ID
Action	Action performed by the firewall (Allow, Deny,Blocked,Etc.)	FRAG	Fragment flags field
IN	Incoming network interface	PROTO	Type of Protocol
OUT	Outgoing interface	SPT	Source port
SRC-IP	Source IP Address	DPT	Destination port
DEST-IP	Destination IP Address	WINDOW	Size of the window
LEN	Length of a packet	RES	Reserved field in the TCP header
TOS	Type of Service field	SYN	TCP state field
		URGP	Urgent pointer

Table 15.8: iptables Log Fields

Example:

```
June 16 21:12:56 FW2 kernel : RULE_08a-ACCEPT IN=eth1 OUT eth0 SRC=192.42.93.30 DST= 192.168.1.102 LEN=96 TOS=0x00 PREC=0x00 TTL=64 ID=61495 DF PROTO=UDP SPT=53981 DPT=127 WINDOW=32767 RES=0x00 SYN URGP=0
```

To enable logging in iptables, the below command is used:

```
$ iptables -A INPUT -j LOG
```

In the above command, source IP or range can be defined in the following manner:

```
$ iptables -A INPUT -s 192.168.10.0/24 -j LOG
```

Similarly, the level of LOG can be defined to generate a specific level of logs:

```
$ iptables -A INPUT -s 192.168.10.0/24 -j LOG --log-level 4
```

Prefixes can be added to search for specific logs in a large file:

```
$ iptables -A INPUT -s 192.168.10.0/24 -j LOG --log-prefix '** SUSPECT **'
```


Monitoring and Analysis of iptables Logs

Use **tail** command for finding recent iptables logs

Execute the command to get the details of recent logs in iptables

```
$ tail -5 /var/log/messages
```

Location of the firewall logs stored

Recent 5 entries of iptables logs

Output:

Timestamp of the log entry	Action Performed	Source IP	Destination IP
Nov 12 12:13:13	kernel: FIREWALL:BLOCKED	SRC=192.168.1.103	DST=192.168.1.255
Nov 12 12:13:13	kernel: FIREWALL:BLOCKED	SRC=192.168.1.103	DST=192.168.1.255
Nov 12 13:22:40	kernel: IPTables-Dropped	SRC=192.168.1.48	DST=192.168.1.255
Nov 12 13:23:00	kernel: IPTables-Dropped	SRC=192.168.2.123	DST=192.168.1.255
Nov 12 21:12:56	FW2 kernel : RULE_0a-ACCEPT	SRC= 192.42.93.30	DST= 192.168.1.255

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of iptables Logs

Linux firewall log helps monitor and analyze incoming and outgoing network traffic. It also enables determining the number of hits from any IP address.

Viewing iptables Log

To view the log, the following command are used (varies with distribution).

- **On Ubuntu and Debian:** To view iptables log in these distributions, the following command can be used: `$ tailf /var/log/kern.log`
- **On CentOS/RHEL and Fedora:** To view iptables log in these distributions, the following command can be used: `# tailf /var/log/messages`

For example, to get the details of recent log records in iptables, execute the following command:

```
$ tail -5 /var/log/messages
```

The above command displays five most recent entries of iptables log.

Cisco ASA Firewall



- Cisco ASA provides advanced application-aware firewall services with identity-based access control and **denial of service** (DoS) attack protection
- Firewall support multiple levels of logging; it helps to address this issue by addressing the most **critical events** first

CISCO Firewall Log Format



1. **Timestamp**: The date and time from the **firewall clock**; the default is no time stamp
2. **Device ID**: Firewall's host name, an interface IP address, or an **arbitrary text string**; the default is no device-id
3. **Message ID**: Begins with %ASA, %PIX, or %FWSM, followed by **severity level** and six-digit message number
4. **Message Text**: Description of the **event** or **condition** that generated the message

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cisco ASA Firewall (Cont'd)



CISCO Firewall Logging Levels

Levels of Logging	Description
Emergencies (0)	System unusable messages
Alerts (1)	Immediate action required messages
Critical (2)	Critical condition messages
Errors (3)	Error condition messages
Warnings (4)	Warning condition messages
Notifications (5)	Normal but significant messages
Informational (6)	Informational messages
Debugging (7)	Debugging messages

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cisco ASA Firewall

Cisco ASA firewall supports multiple levels of logging, which helps to address the issue at hand by prioritizing the most critical events first. These levels of logging are typically labeled 0–7. The logging severity level set for a specific output will not only take logs from that configured severity level but also from all the levels above it. For example, if severity level 7—debugging messages has been configured for the console, then level 7 will not only log all debugging messages but also emergencies, alerts, critical errors, warnings, notifications, and informational messages.

Therefore, always configure the critical severity level for the log messages because setting a higher logging severity level (e.g., 7) would generate a large amount of log messages, which would disturb CPU and memory usage on the Cisco ASA firewall.

The following table depicts the different levels of logging.

Levels of logging	Description
Emergencies (0)	System unusable messages
Alerts (1)	Messages requiring immediate action; for example, failover, power supply, basic RIP, address verification, etc.
Critical (2)	Critical condition messages; for example, denied packets/connections after basic checks, URL filter server problems, etc.
Errors (3)	Error condition messages; for example, authentication/authorization failures, CPU and memory resource problems, tunnel problems, routing and NTP problems, etc.
Warnings (4)	Warning condition messages; for example, fragmentation issues, invalid addresses, auto-update errors, CSPF errors, etc.
Notifications (5)	Normal but significant messages; for example, commands executed by users, configuration events, user and session activity, etc.
Informational (6)	Informational messages; for example, ACL log, authentication/authorization events, firewall startup, fixup activity, etc.
Debugging (7)	Debugging messages; for example, debug messages, TCP/UDP request handling, etc.

Table 15.9: Different Levels of Logging

Cisco Firewall Log Format

Cisco ASA firewall supports two types of format for storing log messages: default and EMBLEM.

- **Default log format:** This type of log format comprises the following types of fields.



Figure 15.27: Cisco Firewall Log Format

- **Time stamp:** It represents the time and date when a log message is generated. It helps in real-time debugging and management, so always add timestamps parameter to log messages. By default, no timestamps are present in the format.

- **Device ID:** It represents firewall's hostname, an interface IP address, or an arbitrary text string. It helps in determining the firewall that produces the log messages. This becomes important when there are multiple firewalls. By default, no device ID is present in the format.
- **Message ID:** It starts with %**ASA**, %**PIX**, or %**FWSM** and is followed by severity level and the six-digit message number.
- **Message text/description:** It mentions the event or condition due to which the log message is generated.
- **EMBLEM log format:** This type of format is mainly used for CiscoWorks Resource Manager Essentials syslog analyzer. This format is similar to Cisco IOS Software syslog format and used by UDP syslog servers only.

Cisco ASA System Log Messages

The following are a few examples of Cisco ASA System log messages:

Mnemonic	Severity	Description
4000 nn ("nn" indicates multiple messages currently 400000–400050)	4	IPS:number string from IP_address to IP_address on interface interface_name
106001	2	Inbound TCP connection denied from IP_address/port to IP_address/port flags tcp_flags on interface interface_name
106002	2	Protocol connection denied by outbound list acl_ID src inside_address dest outside_address
106006	2	Deny inbound UDP from outside_address/outside_port to inside_address/inside_port on interface interface_name
106007	2	Deny inbound UDP from outside_address/outside_port to inside_address/inside_port due to DNS {Response Query}
106010	3	Deny inbound protocol src interface_name:dest_address/dest_port dst
106012	3	Deny IP from IP_address to IP_address, IP options hex
106013	3	Dropping echo request from IP_address to PAT address IP_address
106014	3	Deny inbound icmp src interface_name: IP_address dst interface_name: IP_address (type dec, code dec)

106015	6	Deny TCP (no connection) from IP_address/port to IP_address/port flags tcp_flags on interface interface_name
106016	2	Deny IP spoof from (IP_address) to IP_address on interface interface_name
106017	2	Deny IP due to Land Attack from IP_address to IP_address
106018	2	ICMP packet type ICMP_type denied by outbound list acl_ID src inside_address dest outside_address
106020	2	Deny IP teardrop fragment (size = number, offset = number) from IP_address to IP_address
106021	1	Deny protocol reverse path check from source_address to dest_address on interface interface_name
106022	1	Deny protocol connection spoof from source_address to dest_address on interface interface_name
106023	4	Deny protocol src [interface_name:source_address/source_port] dst interface_name:dest_address/dest_port [type {string}, code {code}] by access_group acl_ID
106100	4	access-list acl_ID {permitted denied est-allowed} protocol interface_name/source_address(source_port) -> interface_name/dest_address(dest_port) hit-cnt number ({first hit number-second interval})
710003	3	{TCP UDP} access denied by ACL from source_IP/source_port to interface_name:dest_IP/service

Table 15.10: Examples of Cisco ASA System Log Messages

Monitoring and Analyzing Cisco ASA Firewall Logs



show logging command generates valuable logs that are **analyzed** to know about the present condition (enabled or disabled) of the device

It helps in **investigating** the state of syslog error, console logging, event logging, monitor logging, etc.

Use **show logging** command with the required (Deny, Outside, Suspicious, etc.) keywords to find the required firewall log messages

grep command, followed by a **regular expression**, yields optimum results

```
Firewall# show logging
Syslog logging: enabled
Facility: 20
Timestamp logging: disabled
Standby logging: disabled
Debug-trace logging: disabled
Console logging: disabled
Monitor logging: enabled
Buffer logging: level informational, 2 messages logged
Trap logging: enabled
Permit-hostdown logging: disabled
History logging: disabled
Device ID: disabled
Mail logging: enabled
ASDM logging: disabled
```

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analyzing Cisco ASA Firewall Logs (Cont'd)



Example: Viewing a Log Entry of Specified Severity by Using the grep Command :

Firewall logging level Access Denied

```
Firewall# show logging | grep ASA-4
Oct 24 2018 08:54:48: %ASA-4-106023: Deny tcp src outside:192.168.208.63/46857
dst inside:192.168.150.77/443 by access-group "OUTSIDE" [0x5063b82f, 0x0]
Oct 24 2018 08:54:48: %ASA-4-106023: Deny tcp src outside:192.168.208.63/46863
dst inside:192.168.150.77/256 by access-group "OUTSIDE" [0x5063b82f, 0x0]
Oct 24 2018 08:54:48: %ASA-4-106023: Deny tcp src outside:192.168.208.63/46867
dst inside:192.168.150.77/389 by access-group "OUTSIDE" [0x5063b82f, 0x0]
```

Destination Address Source Address

- Source ports (46855, 46856, 46857, 46863, and 46867); destination ports (0, 256, 389, and 443)
- The connection from the machine with the IP address 192.168.208.63 is denied access to 192.168.150.77

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analyzing Cisco ASA Firewall Logs

Analyzing log files in Cisco ASA reveals various details that are useful for investigating an incident. Cisco ASA firewall generates a large amount of logging information, only a part of which has importance and should be analyzed. Before analyzing the firewall logs, the important information needs to be determined first.

The following are a few examples of the type of information that is important in firewall logs:

- Connections accepted by firewall rules
- Connections rejected by firewall rules
- User activity
- Bandwidth usage
- Address translation audit trail
- IDS activity
- Protocol usage
- Cut-through proxy activity
- Denied rule rates

Cisco ASA Command Line Interface

Cisco ASA Command Line Interface (CLI) is the console where all the available commands are executed to know the details of the Cisco ASA log. It includes command modes, and some commands can be entered only in specific modes. For example, to enter commands that display confidential information, a password needs to be entered apart from being in a more privileged mode. To enter commands that display configuration change information, the user should be in configuration mode. To enter all lower commands, higher modes need to be accessible; for example, a privileged EXEC command will enter in global configuration mode.

In the system configuration mode or in single context mode, the prompt will start with the hostname:

hostname

If the user is already inside a context, then the prompt will start in the following manner:

hostname/context

Access Modes in Cisco ASA Command Line Interface

Cisco ASA CLI supports four types of access modes: user EXEC mode, privileged EXEC mode, global configuration mode, and command-specific configuration mode. Different modes have different prompt screens.

- **User EXEC mode:** In this mode, basic adaptive security appliance settings can be viewed. Here, the prompt screen will appear in the following way:

hostname>

hostname/context>

- **Privileged EXEC mode:** In this mode, current settings up to the user privilege level can be viewed. The user can run any user EXEC mode command in privileged EXEC mode, and can also switch to privileged EXEC mode in user EXEC mode. To do this, enter `enable` command in user EXEC mode, where a password needs to be provided to switch to the privileged EXEC mode. Here, the prompt screen will appear in the following way:

hostname#

hostname/context#

- **Global configuration mode:** This mode enables changes in the adaptive security appliance configuration. This mode contains all users EXEC, privileged EXEC, and global configuration commands. A user can switch to global configuration mode from privileged EXEC mode by entering **configure** terminal command. Here, the prompt screen will appear in the following way:

hostname (config) #

hostname/context (config) #

- **Command-specific configuration mode:** This mode includes commands from user EXEC mode, privileged EXEC mode, global configuration mode, and command-specific configuration mode. Here, a prompt screen will appear in the following way:

hostname (config-if) #

hostname/context (config-if) #

Viewing ASA Firewall Logging

The **show logging** command is used to view Cisco ASA firewall logs. It generates a variety of valuable log entries that are analyzed to know the present condition (enabled or disabled) of the device. It also helps in investigating the state of syslog error, console logging, event logging, monitor logging, etc. It also displays the syslog message that begins with %ASA followed by the logging level, the message ID, and a brief description of the log message.

Example:

```
Firewall# show logging
Syslog logging: enabled
Facility: 20
Timestamp logging: disabled
Standby logging: disabled
Debug-trace logging: disabled
Console logging: disabled
Monitor logging: enabled
Buffer logging: level informational, 2 messages logged
Trap logging: enabled
Permit-hostdown logging: disabled
History logging: disabled
Device ID: disabled
Mail logging: enabled
ASDM logging: disabled
```

In the above example, the **show logging** command displays different pieces of information, which convey that: logging is enabled globally, timestamps logging is disabled, and console logging is also disabled, may be it would be on production devices. Information regarding the total number of logged messages for each configured destination can also be known in similar manner.

Filtering show Command Output

The **show logging** command can be used with various keywords such as **include**, **exclude**, **begin**, **grep**, etc. to find the required firewall log messages. The included keywords display information that matches the regular expression, and the **grep** keyword without **-v** has the same action. The **exclude** keyword excludes information that matches the regular expression, and the **grep** keyword with **-v** has the same action. The **begin** option displays the information beginning the line that matched the regular expression. Among the various keywords, **grep** command followed by a regular expression will yield optimum results. **grep** command can also be used to fetch log messages of a specific severity.

For example, to view firewall log records of a specific severity through the **grep** command:

```
Firewall# show logging | grep ASA-4
```

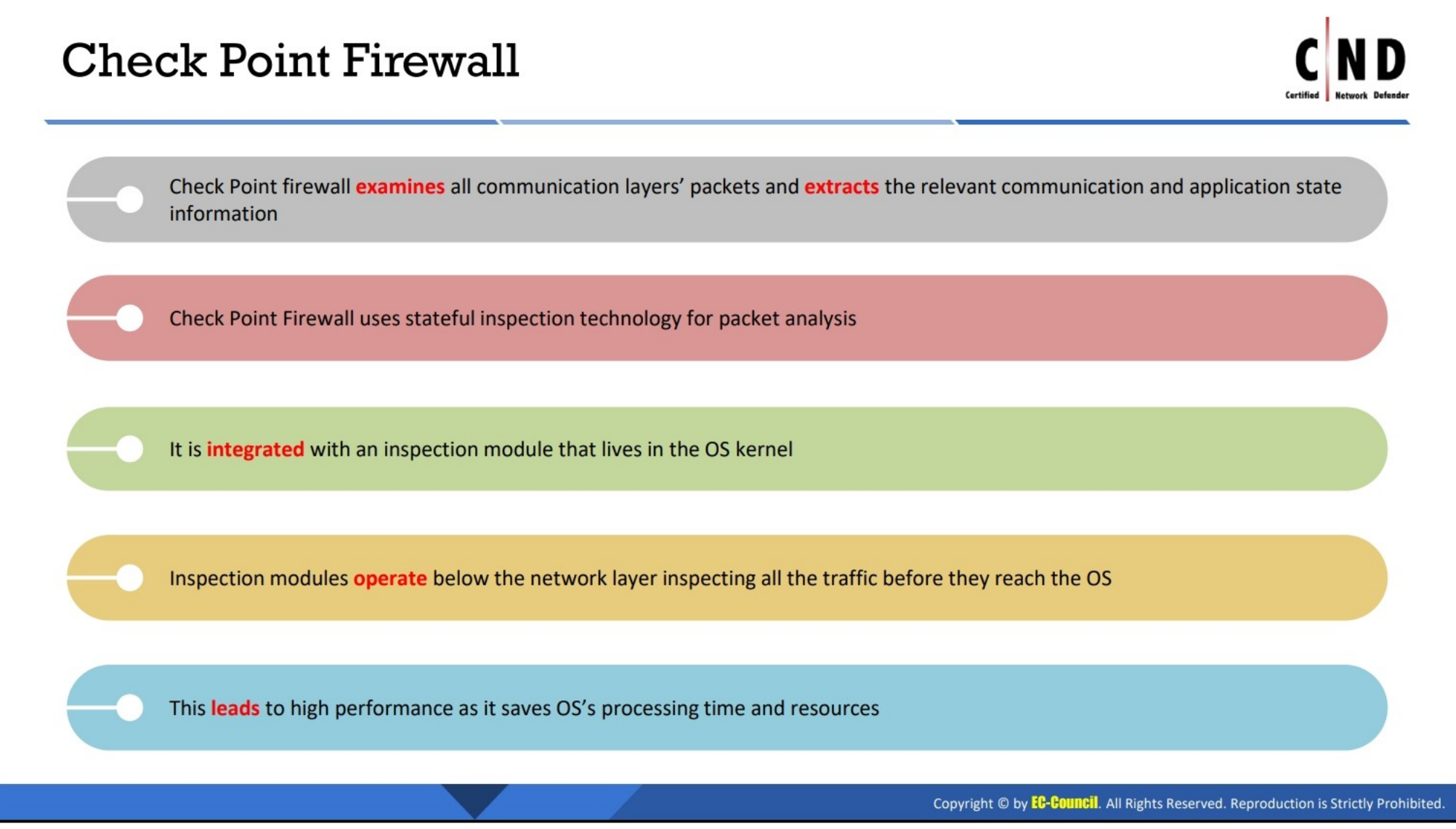
```
Oct 24 2018 08:54:48: %ASA-4-106023: Deny tcp src  
outside:192.168.208.63/46857  
dst inside:192.168.150.77/443 by access-group "OUTSIDE" [0x5063b82f,  
0x0]
```

```
Oct 24 2018 08:54:48: %ASA-4-106023: Deny tcp src  
outside:192.168.208.63/46863  
dst inside:192.168.150.77/256 by access-group "OUTSIDE" [0x5063b82f,  
0x0]
```

```
Oct 24 2018 08:54:48: %ASA-4-106023: Deny tcp src  
outside:192.168.208.63/46867  
dst inside:192.168.150.77/389 by access-group "OUTSIDE" [0x5063b82f,  
0x0]
```

The above example shows information regarding ASA with severity level 4 where source ports are 46857, 46863, and 46867 and destination ports are 256, 389, and 443. It also shows that the IP address 192.168.208.63 was denied access to 192.168.150.77.

Check Point Firewall

An infographic titled "Check Point Firewall" with the CND logo in the top right. It features five horizontal bars of different colors (grey, red, green, yellow, blue) each containing a bullet point. The text in the bars describes the firewall's operation: examining all communication layers' packets and extracting relevant information; using stateful inspection technology; being integrated with an OS kernel inspection module; operating below the network layer; and leading to high performance by saving OS resources. A copyright notice for EC-Council is at the bottom right of the infographic.

- Check Point firewall **examines** all communication layers' packets and **extracts** the relevant communication and application state information
- Check Point Firewall uses stateful inspection technology for packet analysis
- It is **integrated** with an inspection module that lives in the OS kernel
- Inspection modules **operate** below the network layer inspecting all the traffic before they reach the OS
- This **leads** to high performance as it saves OS's processing time and resources

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Check Point Firewall

Check Point firewall examines all communication layers' packets and extracts the relevant communication and application state information. It uses stateful inspection technology for packet analysis, where incoming packets are inspected intelligently and the packets that seem to be dangerous are blocked. Check Point firewall is integrated with an inspection module that lives in the OS kernel. The inspection module operates below the network layer, inspecting each packet and ensuring that it will not enter in a network until it complies with the network's security policy.

Check Point firewall is more advanced when compared to traditional firewalls. Traditional firewalls analyze only the message headers, but Check Point firewall inspects the complete raw message and analyzes all data from packet communication layers. Initially, it inspects the IP addresses, port numbers, and other information to identify whether packets are according to the network security policy. Then, it verifies whether the connection is coming from the appropriate source or not. For this, it has to match the incoming packet against state and context information, which is located in dynamic static tables. These are tables that store cumulative data through which Check Point firewall checks upcoming transmissions. If the incoming traffic does not match with the network standards and policy, then the firewall generates real-time alerts to the network defenders.

Monitoring and Analyzing Check Point Firewall Logs



fw log command is used to display the log file content

Syntax:

fw log [-f [-t]] [-n] [-l] [-o] [-c action] [-h host] [-s starttime] [-e endtime] [-b starttime endtime] [-u unification_scheme_file] [-m unification_mode(initial|semi|raw)] [-a] [-k (alert_name|all)] [-g] [logfile]

Table with 2 columns: Parameter, Description. Rows include -f [-t], -n, -l, -o, -c action, -h host, -s starttime, -e endtime, -b starttime endtime.

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analyzing Check Point Firewall Logs (Cont'd)



Table with 2 columns: Parameter, Description. Rows include -u unification_scheme_file, -m unification_mode, -a, -k alert_name, -g, logfile, -u unification_scheme_file, -m unification_mode, -a.

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analyzing Check Point Firewall Logs (Cont'd)



Each line of **fw log** command's output represents a single record; each field of log appears in the following format:

[<date>] <time> <action> <origin> <interface dir and name> [alert] [field name: field value;] ...

fw log Example

```
15:56:39; reject dam.checkpoint.com >daemon alert; src: veredr.checkpoint.com; dst: dam.checkpoint.com; user: a; rule: 0; reason: Client Encryption: Access denied - wrong user name or password; scheme: IKE; reject_category: Authentication error; product: Security Gateway
15:57:49; authcrypt dam.checkpoint.com >daemon; src: veredr.checkpoint.com; user: a; rule: 0; reason: Client Encryption: Authenticated by Internal Password; scheme: IKE; methods: AES-256, IKE, SHA1; product: Security Gateway;
15:57:49; keyinst dam.checkpoint.com >daemon; src: veredr.checkpoint.com; peer_gateway: veredr.checkpoint.com; scheme: IKE; IKE: Main Mode completion.; CookieI: 32f09ca38aeaf4a3; CookieR: 73b91d59b378958c; msgid: 47ad4a8d; methods: AES-256 + SHA1, Internal Password; user: a; product: Security Gateway;
```

1 Time

2 Action

3 Origin

4 Interface directory and name

5 Alert

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analyzing Check Point Firewall Logs

Check Point firewall log records can be viewed via Check Point GUI by issuing the **fw log** command. The log ins/log under can be found in the installation directory—**\$FWDIR**. Check Point firewall log records can also be viewed via CLI. For this, first connect to the Check Point firewall platform through SSH or a console over a TCP/IP network and then enter the credentials to log into the CLI.

Check Point firewall log contains traffic log and audit log entries, as described below.

- **Traffic logs:** These are the most useful log entries in the main log. They represent traffic that is allowed, dropped, or denied. These log entries are beneficial for detecting port scans, host sweeps, and general probing. Check Point firewall generates accept alerts when network traffic is allowed and deny or drop alerts when network traffic is not allowed. These alerts include a rule that helps in troubleshooting issues.
- **Audit logs:** These log entries reflect all changes performed through the GUI. Each log entry represents the user who logged in, the machine name from where they came from, the component they used, the authentication technique they applied, and the change performed. These log entries are useful for general auditing and analyzing a compromised firewall host.

The fw log Command

The **fw log** command is used to display the content of the Check Point log file. Logs can also be viewed in real time by using the **fw log -ftn** command. This command is used by various network defenders to send Check Point logs securely over the network. The syntax of this command is given below:


```
fw log [-f [-t]] [-n] [-l] [-o] [-c action] [-h host] [-s starttime] [-e endtime] [-b starttime endtime] [-u unification_scheme_file] [-m unification_mode(initial|semi|raw)] [-a] [-k (alert_name|all)] [-g] [logfile]
```

Each line of `fw log` command's output represents a single record, and each field of the log appears in the following format:

```
[<date>] <time> <action> <origin> <interface dir and name> [alert] [field name: field value;] ...
```

The various fields used in the above syntax are described below.

- **date:** Date on which action is generated in the MMM DD, YYYY format; for example, Feb 16, 2018
- **time:** Time at which action is generated in the HH:MM:SS format; for example, 15:22:00
- **action:** Action performed by the firewall such as accept, drop, reject, authorize, deauthorize, encrypt, and decrypt
- **origin:** Firewall that wrote the record
- **interface dir:** Firewall interface directory
- **interface name:** Firewall interface name
- **alert:** Type of alert generated
- **field name:** Name of the other field name
- **field value:** Specified field value

Example:

```
15:56:39      reject      dam.checkpoint.com      >daemon      alert      src: veredr.checkpoint.com; dst: dam.checkpoint.com; user: a; rule: 0; reason: Client Encryption: Access denied - wrong username or password; scheme: IKE; reject_category: Authentication error; product: Security Gateway
```

```
15:57:49      authcrypt      dam.checkpoint.com      >daemon      src: veredr.checkpoint.com; user: a; rule: 0; reason: Client Encryption: Authenticated by Internal Password; scheme: IKE; methods: AES-256,IKE,SHA1; product: Security Gateway;
```

```
15:57:49 keyinst dam.checkpoint.com >daemon src: veredr.checkpoint.com; peer gateway: veredr.checkpoint.com; scheme: IKE; IKE: Main Mode completion.; CookieI: 32f09ca38aeaf4a3; CookieR: 73b91d59b378958c; msgid: 47ad4a8d; methods: AES-256 + SHA1, Internal Password; user: a; product: Security Gateway;
```

The above examples display varieties of information such as time (15:56:39, 15:57:49), action (reject dam.checkpoint.com >daemon alert, authcrypt dam.checkpoint.com >daemon, keyinst dam.checkpoint.com >daemon), origin (src: veredr.checkpoint.com; src: veredr.checkpoint.com; src: veredr.checkpoint.com), interface directory and name (user: a, user: a, peer gateway:

veredr.checkpoint.com), and alert (reason: Client Encryption: Access denied—wrong username or password).

Check Point Firewall-Specific Logging Issues and Challenges

- Logs generated by Check Point firewall are not in a readable format.
- GUI log viewer does not provide real-time log analysis view. For this, you have to look at checkpoint devices.
- It is not beneficial for batch analysis.
- It is limited to filtering and sorting.



LO#06: Discuss log monitoring and analysis on routers

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

LO#06: Log Monitoring and Analysis on Routers

The objective of this section is to explain how to monitor and analyze router logs. Specifically, it demonstrates how to monitor and analyze Cisco router logs.

Cisco Router Log



- Router log messages do not contain **numerical identifiers** that assist in identifying the messages
- It includes a maximum of 80 characters and a percentage sign (%), followed by an optional sequence number or timestamp information if configured

Router Log Messages that are Most Useful When Analyzing Security-Related Incidents

Mnemonic	Severity	Description
%SEC-6-IPACCESSLOGDP	6	A packet matching the log criteria for the given access list has been detected
%SEC-6-IPACCESSLOGNP	6	A packet matching the log criteria for the given access list has been detected
%SEC-6-IPACCESSLOGP	6	A packet matching the log criteria for the given access list has been detected (TCP OR UDP)
%SEC-6-IPACCESSLOGRL	6	Some packet-matching logs were missed because the access 1st log messages were rate limited, or no access 1st log buffers were available
%SEC-6-IPACCESSLOGRP	6	A packet matching the log criteria for the given access list has been detected
%SEC-6-IPACCESSLOGS	6	A packet matching the log criteria for the given access list has been detected
%SEC-4-TOOMANY	6	A packet matching the log criteria for the given access list has been detected
%SEC-6-IPACCESSLOGP	6	A packet matching the log criteria for the given access list has been detected
%SEC-6-IPACCESSLOGDP	6	A packet matching the log criteria for the given access list has been detected
%SEC-6-IPACCESSLOGNP	6	A packet matching the log criteria for the given access list has been detected

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Cisco Router Log

Router log messages play an important role in analyzing security-related incidents. They include a maximum of 80 characters and a percent sign (%), followed by an optional sequence number or timestamp information (if configured). They do not contain numerical identifiers that assist in identifying the messages. Router log messages are represented in the following format:

seq no:timestamp: %facility-severity-MNEMONIC:description

The various fields used in the above syntax are described below.

- seq no:** This field represents stamps log messages with a sequence number. This information will be displayed only if the **service sequence-numbers** global configuration command is configured.
- timestamp:** This field represents the date and time of the message on which it is generated. The date and time is in the mm/dd hh:mm:ss, hh:mm:ss (short uptime), or d h (long uptime) format. This information will be displayed only if the **service timestamps log [datetime | log]** global configuration command is configured.
- facility:** This field represents the facility. It may be a hardware device, a protocol, or a module. It determines the source and the reason why the message is generated.
- severity:** This field represents the severity level of the message from 0 to 7.
- MNEMONIC:** It is a text string that describes the message uniquely.
- description:** This field represents detailed information regarding the event generated.

The following table shows the examples of mnemonics use in Cisco router log:

Mnemonic	Severity	Description
%SEC-6-IPACCESLOGDP	6	A packet matching the log criteria for the given access list has been detected
%SEC-6-IPACCESLOGNP	6	A packet matching the log criteria for the given access list has been detected
%SEC-6-IPACCESLOGP	6	A packet matching the log criteria for the given access list has been detected (TCP OR UDP)
%SEC-6-IPACCESLOGRL	6	Some packet-matching logs were missed because the access first log messages were rate limited, or no access first log buffers were available
%SEC-6-IPACCESLOGRP	6	A packet matching the log criteria for the given access list has been detected
%SEC-6-IPACCESLOGS	6	A packet matching the log criteria for the given access list has been detected
%SEC-4-TOOMANY	6	A packet matching the log criteria for the given access list has been detected
%SEC-6-IPACCESLOGP	6	A packet matching the log criteria for the given access list has been detected
%SEC-6-IPACCESLOGDP	6	A packet matching the log criteria for the given access list has been detected
%SEC-6-IPACCESLOGNP	6	A packet matching the log criteria for the given access list has been detected

Table 15.11: Examples of Cisco Router Log Mnemonics

Severity Levels of Cisco Router Logs

Log messages in Cisco routers are categorized into eight severity levels ranging from 0 to 7. Each severity level is given a number and its corresponding name and UNIX syslog definitions. The lower severity number represents a higher severity and vice-versa.

Level	Level name	Syslog definition	Description
0	Emergencies	LOG_EMERG	System unusable
1	Alerts	LOG_ALERT	Immediate action needed
2	Critical	LOG_CRIT	Critical conditions
3	Errors	LOG_ERR	Error conditions
4	Warnings	LOG_WARNING	Warning conditions

5	Notifications	LOG_NOTICE	Normal but significant
6	Informational	LOG_INFO	Informational messages
7	Debugging	LOG_DEBUG	Debugging messages

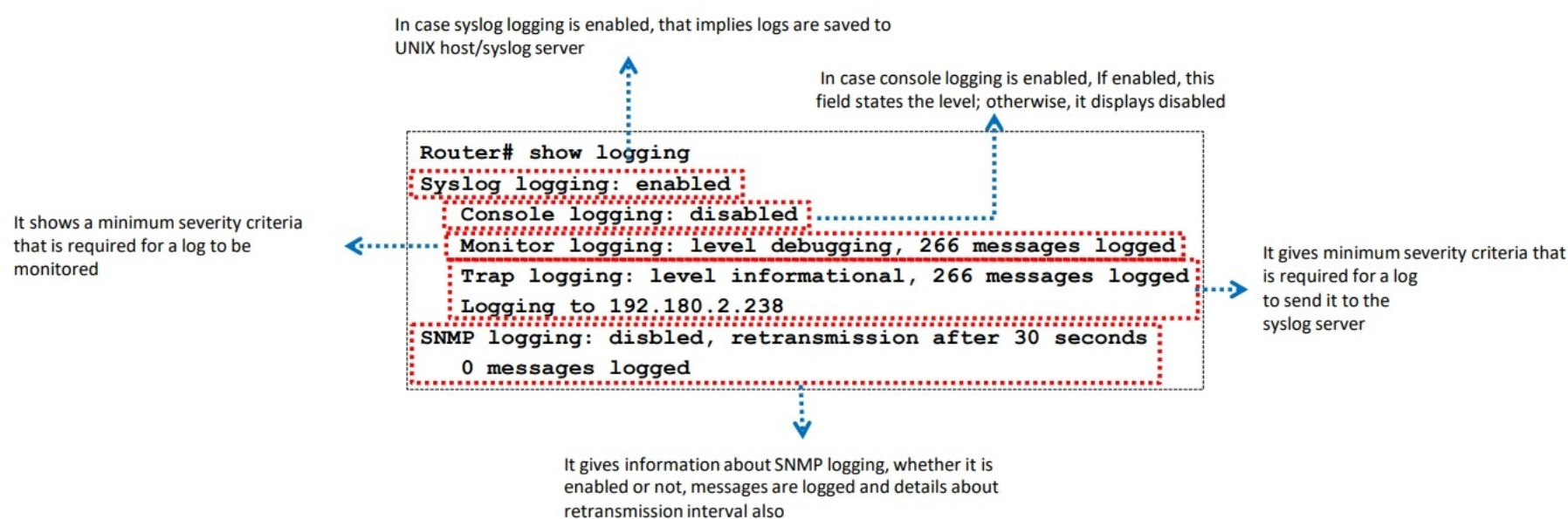
Table 15.12: Severity levels of Cisco Router Logs

- Messages between warning and emergency levels are designated as error messages that represent software and hardware malfunctions.
- Interface up or down transition messages, system restart messages, and other informational messages are shown as notifications.
- Information messages represent reload requests, low-process stack messages, etc.
- Debug-level messages represent outputs provided by the debug commands.

Monitoring and Analysis of Router Logs



- **show logging** helps in **investigating** the state of syslog error, console logging, event logging, and host addresses
- It helps in finding to what **levels** various outputs are set and where ultimately output is sent



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Router Logs (Cont'd)



- Use **show logging** command with **include** filter to search for specific keywords in the router logs

```
Router# show logging | include 185
002092: Dec 28 2018 12:20:48.681 EDT: %SEC-6-IPACCESSLOGP: list 185 denied tcp 168.215.25.92(59078) ->
71.15.210.4(417), 1 packet
002093: Dec 28 2018 12:20:49.681 EDT: %SEC-6-IPACCESSLOGP: list 185 denied tcp 168.215.25.95(14897) ->
71.15.210.4(427), 1 packet
002094: Dec 28 2018 12:20:50.681 EDT: %SEC-6-IPACCESSLOGP: list 185 denied tcp 168.215.25.182(16737)->
71.15.210.4(437), 1 packet
002095: Dec 28 2018 12:20:56.985 EDT: %SEC-6-IPACCESSLOGP: list 185 denied tcp 168.215.25.219(14872) (FastEthernet0/1
0007.8580.9edd) -> 71.15.210.4(500), 1 packet
002096: Dec 28 2018 12:20:57.984 EDT: %SEC-6-IPACCESSLOGP: list 185 denied tcp 168.215.25.208(7751) (FastEthernet0/1
0007.8580.9edd) -> 71.15.210.4(510), 1 packet
002097: Dec 28 2018 12:20:58.984 EDT: %SEC-6-IPACCESSLOGP: list 185 denied tcp 168.215.25.26(41202) (FastEthernet0/1
0007.8580.9edd) -> 71.15.210.4(520), 1 packet
Router#
```

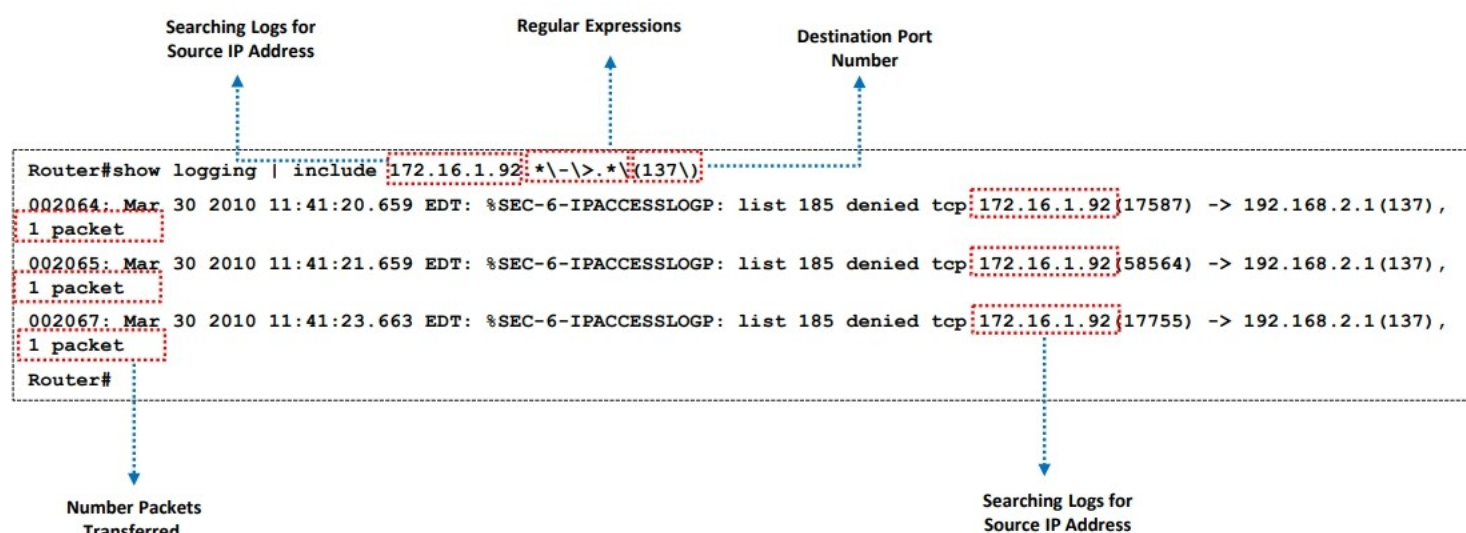
- In this example, **show logging | include 185** displays all the logs generated by the access control list 185

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Router Logs (Cont'd)



Use **include** command with regular expressions for identifying intrusions



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Router Logs

The **show logging** command helps in investigating the state of syslog error, console logging, event logging, and host addresses. It also shows configuration parameters and protocol activity of SNMP logging. Further, it can display information regarding the standard system logging buffer (only if **logging buffered** command is enabled). You can also configure the size of the syslog buffer using the **logging buffered** command. This determines the number of system error and debugging messages to be stored in the system logging buffer. The following example shows a sample output from the **show logging** command:

Router# show logging

Syslog logging: enabled

Console logging: disabled

Monitor logging: level debugging, 266 messages logged

Trap logging: level informational, 266 messages logged

Logging to 192.180.2.238

SNMP logging: disabled, retransmission after 30 seconds

0 messages logged

The following table describes the various fields used in the above output:

Field	Description
Syslog logging	In case syslog logging is enabled, it implies that logs are saved to UNIX host/syslog server.
Console logging	In case console logging is enabled, this field states the level; otherwise, it displays disabled.
Monitor logging	It shows minimum severity criteria that are required for a log to be monitored.
Trap logging	It gives minimum severity criteria that are required for a log to send it to the syslog server.
SNMP logging	It gives information about SNMP logging, whether it is enabled or not, whether messages are logged, and details about retransmission interval as well.

Table 15.13: Number of Fields and its Description

The **show logging** command can combine the keywords with search-specific filters to identify relevant information. The **history** keyword can be used to fetch information regarding the syslog history table in the following manner:

```
Router# show logging history
```

```
Syslog History Table: 1 maximum table entry, saving level notifications or higher
```

```
0 messages ignored, 0 dropped, 15 table entries flushed,
```

```
SNMP notifications not enabled
```

```
entry number 16: SYS-5-CONFIG_I
```

```
Configured from console by console
```

```
timestamp: 1110
```

The following table represents the information specified in the above sample:

Fields	Description
Maximum table entry	It represents how many messages can be stored in the history table. This is configured by using the logging history size command.
Saving level notifications or higher	It describes the level up to which the messages can be stored in the history table. This is configured by using logging history size command.
Messages ignored	It describes how many messages are not stored in the history table.

Dropped	It describes how many messages are not processed due to insufficient resources.
Table entries flushed	It describes how many messages are deleted from the history table to store new messages.
SNMP notifications	It describes whether syslog traps are sent to the SNMP server. Use snmp-server command to enable syslog traps.
Entry number	It describes how many messages are there in the history table.
SYS-5-CONFIG_I Configured from console by console	It describes the Cisco IOS syslog message.
Timestamp	It describes the time at which message was generated.

Table 15.14: Types of Information and their Description

The **include** keyword with **show logging** command can be used to search for specific keywords in the router logs. See below for an example:

```
Router# show logging | include 185
```

```
002092: Dec 28 2018 12:20:48.681 EDT: %SEC-6-IPACCESSLOGP: list 185
denied tcp 168.215.25.92(59078) -> 71.15.210.4(417), 1 packet
002093: Dec 28 2018 12:20:49.681 EDT: %SEC-6-IPACCESSLOGP: list 185
denied tcp 168.215.25.95(14897) -> 71.15.210.4(427), 1 packet
002094: Dec 28 2018 12:20:50.681 EDT: %SEC-6-IPACCESSLOGP: list 185
denied tcp 168.215.25.182(16737) -> 71.15.210.4(437), 1 packet
002095: Dec 28 2018 12:20:56.985 EDT: %SEC-6-IPACCESSLOGP: list 185
denied tcp 168.215.25.219(14872) (FastEthernet0/1 0007.8580.9edd) ->
71.15.210.4(500), 1 packet
002096: Dec 28 2018 12:20:57.984 EDT: %SEC-6-IPACCESSLOGP: list 185
denied tcp 168.215.25.208(7751) (FastEthernet0/1 0007.8580.9edd) ->
71.15.210.4(510), 1 packet
002097: Dec 28 2018 12:20:58.984 EDT: %SEC-6-IPACCESSLOGP: list 185
denied tcp 168.215.25.26(41202) (FastEthernet0/1 0007.8580.9edd) ->
71.15.210.4(520), 1 packet
```

```
Router#
```

In the above example, **show logging | include 185** displays all the logs generated by the access control list 185.

The **include** command can also be used with regular expressions for identifying intrusions. See below for an example:

```
Router#show logging | include 172.16.1.92.*\->.*\ (137\)
```



```
002064: Mar 30 2018 11:41:20.659 EDT: %SEC-6-IPACCESSLOGP: list 185
denied tcp 172.16.1.92(17587) -> 192.168.2.1(137), 1 packet
```

```
002065: Mar 30 2018 11:41:21.659 EDT: %SEC-6-IPACCESSLOGP: list 185
denied tcp 172.16.1.92(58564) -> 192.168.2.1(137), 1 packet
```

```
002067: Mar 30 2018 11:41:23.663 EDT: %SEC-6-IPACCESSLOGP: list 185
denied tcp 172.16.1.92(17755) -> 192.168.2.1(137), 1 packet
```

Router#

In the above example, `show logging | include 172.16.1.92.*\->.*\ (137\)` will display all the log entries with source IP address 172.16.1.92 and destination port 137.




LO#07: Discuss log monitoring and analysis on web servers


Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

LO#07: Log Monitoring and Analysis on Web Servers


The objective of this section is to explain how to monitor and analyze logs in a web server. Specifically, it demonstrates how to monitor and analyze Internet Information Services (IIS) and Apache logs.

Internet Information Services (IIS) Logs







Internet Information Services (IIS) is a **web server for Windows server** that hosts anything on the Web



IIS consists of many log files; log file formats provide different information of the users IP address, different sites visited by the user with date and time



IIS log file provides useful information regarding the person who visited your site, what information was viewed and when it was viewed, the activity of various web applications, etc.



Proper analysis of IIS log files provides **demographic information** and **usage of IIS server**

■ The log files are located by default at:

IIS 6.0
`%system32%\LogFiles\W3SVCN`

IIS 7.0
`%SystemDrive%\Inetpub\Logs\LogFiles\W3SVCN`

IIS 8.0
`%SystemDrive%\inetpub\logs\LogFiles`

IIS 10.0
`%SystemDrive%\inetpub\logs\LogFiles`

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Internet Information Services (IIS) Logs

Internet Information Services (IIS) consists of many log files that provide useful information regarding the person who visited the hosted website, what information was viewed and when it was viewed, the activity of various web applications, etc. Proper analysis of IIS log files will also provide demographic information and usage of the IIS server. By monitoring data usage, web providers can effectively organize their services to support specific regions, time frames, or IP ranges. Log filters also facilitate providers to determine only that specific data that is required for analysis.

By default, IIS log files are located in `%SystemDrive%\inetpub\logs\LogFiles` in IIS 8.0 and 10.0, `%SystemDrive%\Inetpub\Logs\LogFiles\W3SVCN` in IIS 7.0, and `%system32%\LogFiles\W3SVCN` in IIS 6.0. If they are not located in the default location, then the following steps are used to determine their location:

- Open IIS Manager.

- Double click on the Logging icon that appears in the middle pane under the section IIS.

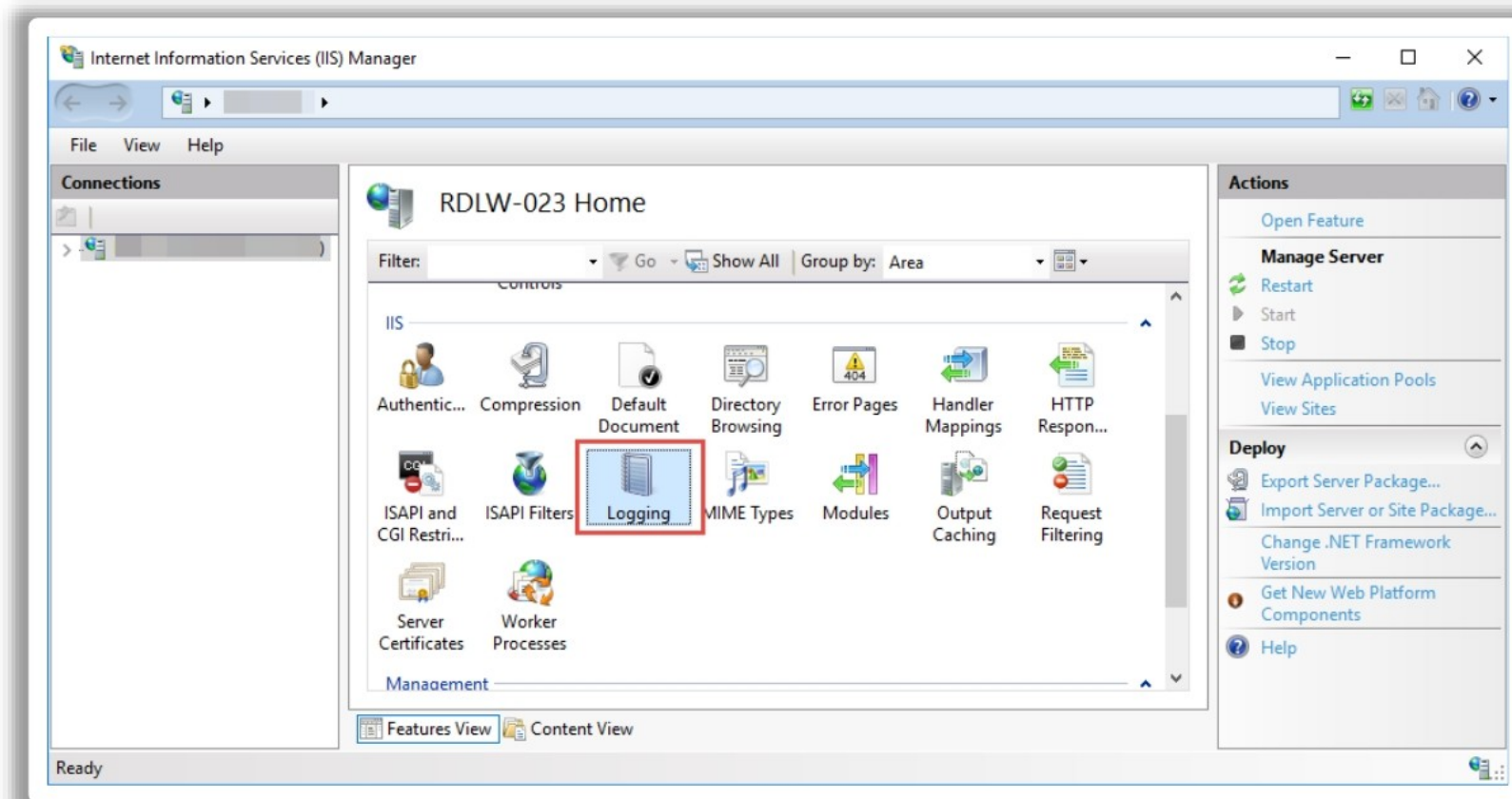


Figure 15.28: Screenshot of IIS Manager

- Logging setting screen will appear, where you can find the location of the IIS log files under the Directory field.

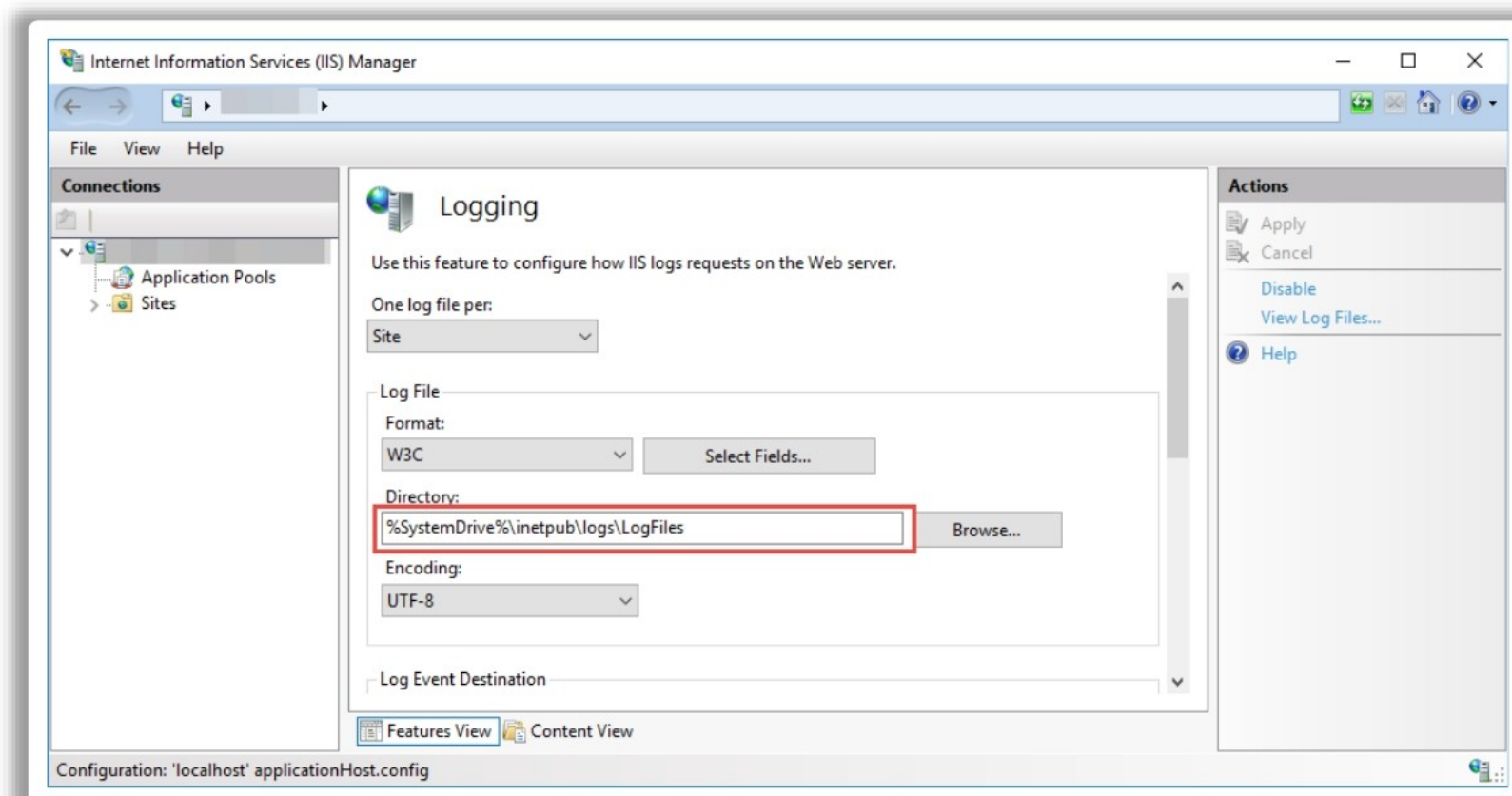


Figure 15.29: Screenshot of "Logging" Setting Screen

- Navigate to the IIS log files location mentioned in the Directory field. The folders store the log files having a naming pattern such as W3SVC1, W3SVC2, etc.

IIS Log File Format

IIS logs keep records of site activity in different formats such as the W3C Extended log file format, IIS log file format, and National Center for Supercomputing Applications (NCSA) Common log file format. All these log file formats are ASCII text formats. In NCSA and IIS log format, logged data is fixed for each request. However, in W3C Extended log format, logged data is not fixed; instead, different properties can be selected for each request. Different log formats use different time

zones to determine when a specific event is generated. W3C Extended format uses Coordinated Universal Time (UTC) whereas other formats use local time.

- **W3C Extended log file format:** It is a customizable ASCII format with different properties, separated with spaces. This format enables removal of unwanted property fields to limit the log size. Here, log records are recorded in UTC time zone. The below example shows W3C Extended log file format with properties such as time, client IP address, method, URI stem, protocol status, and protocol version.

```
#Software: Internet Information Services 10.0
```

```
#Version: 1.0
```

```
#Date: 2019-05-02 17:42:15
```

```
#Fields: time c-ip cs-method cs-uri-stem sc-status cs-version  
17:42:15 172.16.255.255 GET /default.htm 200 HTTP/1.0
```

The W3C Extended log file format contains the following types of fields:

Field name	Description	Uses
Date (date)	The date of the request	Event correlation
Time (time)	The UTC time of the request	Event correlation, determine time zone
Client IP address (c-ip)	The IP address of the client or proxy that sent the request	Identify user or proxy server
Username (cs-username)	The username used to authenticate to the resource	Identify compromised user passwords
Service name (s-sitename)	The W3SVC instance number of the site accessed	Can verify the site accessed if the log files are later moved from the system
Server name (s-computername)	Windows hostname assigned to the system that generated the log entry	Can verify the server accessed if the log files are later moved from the system
Server IP address (s-ip)	The IP address that received the request	Can verify the IP address accessed if the log files are later moved from the system or if the server is moved to a new location

Server port (s-port)	The TCP port that received the request	The TCP port that received the request will verify the port when correlating with other types of request log files
Method (cs-method)	The HTTP method used by the client	Can help track down abuse of scripts or executables
URI stem (cs-uri-stem)	The resource accessed on the server	Can identify attack vectors
URI query (cs-uri-query)	The contents of the query string portion of the URI	Can identify injection of malicious data
Protocol status (sc-status)	The result code sent to the client	Can identify CGI scans, SQL injection, and other intrusions
Win32 status (sc-win32-status)	The win32 error code produced by the request	Can help identify script abuse
Bytes sent (sc-bytes)	The number of bytes sent to the client	Can help identify unusual traffic from a single script
Bytes received (cs-bytes)	The number of bytes received from the client	Can help identify unusual traffic from a single script
Time taken (time-taken)	The amount of server time, in milliseconds, taken to process the request	Can help identify unusual traffic from a single script
Protocol version (cs-version)	The HTTP protocol version supplied by the client	Can help identify older scripts or browsers
Host (cs-host)	The contents of the HTTP host header sent by the client	Can determine if the user browsed to the site by IP address or hostname
User agent (cs (User-Agent))	The contents of the HTTP user agent header sent by the client	Can help uniquely identify users or attack scripts
Cookie (cs (Cookie))	The contents of the HTTP cookie header sent by the client	Can help uniquely identify users
Referer (cs (Referer))	The contents of the HTTP referer header sent by the client	Can help identify the source of an attack or see if an attacker is using search

		engines to find vulnerable sites
--	--	----------------------------------

Table 15.15: Types of Fields in “W3C Extended” Log File Format

- **IIS log file format:** It is a fixed (cannot be customized) ASCII text-based format. It records more information as compared to the NCSA Common format. This format includes basic items such as client IP address, user information, date and time, service and instance, service status code, server name, and IP address, request type, number of bytes received, number of bytes sent, the target of operation, etc. Here, each item is separated by comma and it uses local time to record time. The below example shows an IIS log file entry in the IIS log file format:

```
192.168.100.150, -, 03/6/11, 8:45:30, W3SVC2, SERVER,
172.15.10.30, 4210, 125, 3524, 100, 0, GET, /dollerlogo.gif, -,
```

These values are represented in the following table along with the corresponding field values:

Field	Appear as	Description
Client IP address	192.168.100.150	IP address of the client
Username	–	User is anonymous
Date	03/06/2011	Log file entry was made on June 03, 2011
Time	8:45:30	Log life entry was recorded at 8:45 A.M.
Service and instance	W3SVC2	This is a website, and the site instance is 2
Server name	SERVER	Name of the server
Server IP	172.15.10.30	IP address of the server
Time taken	4210	This action took 4,210 ms
Client bytes sent	125	Number of bytes sent from client to server
Server bytes sent	3524	Number of bytes sent from server to client
Service status code	100	Request was fulfilled successfully
Windows status code	0	Request was fulfilled successfully

Request type	GET	User issued a GET or download command
Target of operation	/dollerlogo.gif	User wanted to download the DeptLogo.gif file
Parameters	–	No parameters passed

Table 15.16: Types of Fields in IIS Log File Format

- **NCSA Common log file format:** Similar to the IIS log file format, it is also is a fixed ASCII text-based format. However, it is used for websites and not for FTP sites. It records items related to user requests such as remote hostname, username, date, time, request type, HTTP status code, and the number of bytes sent by the server. Here, each item is separated by spaces, and it record time based on the local time. The below example shows IIS log file entry in the NCSA log file format in the following manner:

```
13.45 - Microsoft\fred [08/Apr/2001:17:39:04 -0800] "GET
/scripts/iisadmin/ism.dll?http/serv HTTP/1.0" 200 3401
```


Monitoring and Analyzing Log Files in IIS



- Open the log file in the text editor; the **six digits** of the log file name represents the day, month, and year when the file was created. (e.g., "u_ex180405.log")
- Trace the header information line that starts with **"#Fields:"**; this line is used to determine the corresponding values of each column
- Identify when the request was created with the date and time, "sitename" and "computername" indicate which server responded to the request
- Identify who visited the web server with **"c-ip"** (visitor computers IP address)
- The cs-method column contains "post" or "get" requests made by the visitor's browser; "cs-uri-stem" and "cs-uri-query" represent the resource (image/website) requested by the visitor
- Use **"sc-status"** column to find out the capability of the server responding to the request
- Use **"cs(User-Agent)"** to find out which type of browser is used by the visitor

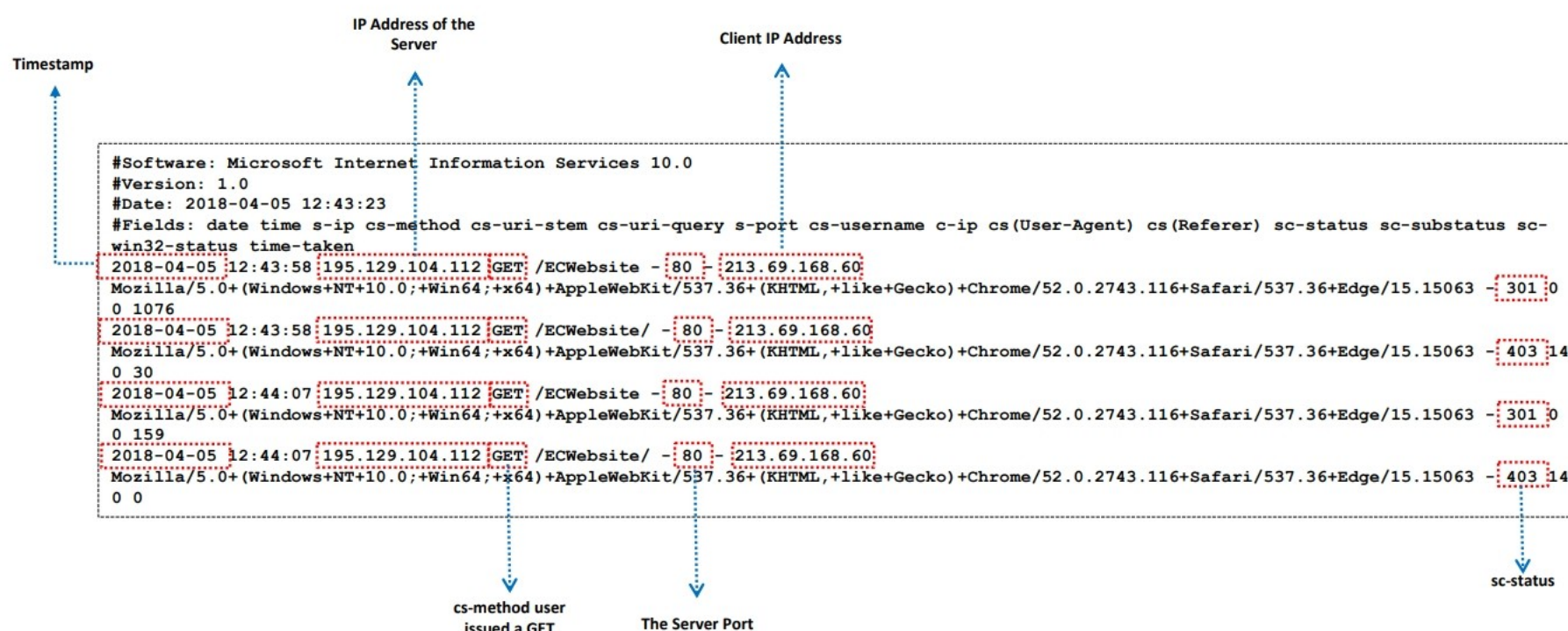
```

1 #Software: Microsoft Internet Information Services 10.0
2 #Version: 1.0
3 #Date: 2018-04-05 12:43:23
4 #Fields: date time s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs(User-Agent) cs(Referer) sc-status sc-substatus sc-win32-status time-taken
5 2018-04-05 12:43:58 195.129.104.112 GET /ECWebsite/ - 80 - 213.69.168.60 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/52
6 2018-04-05 12:43:58 195.129.104.112 GET /ECWebsite/ - 80 - 213.69.168.60 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/52
7 2018-04-05 12:44:07 195.129.104.112 GET /ECWebsite/ - 80 - 213.69.168.60 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/52
8 2018-04-05 12:44:07 195.129.104.112 GET /ECWebsite/ - 80 - 213.69.168.60 Mozilla/5.0+(Windows+NT+10.0;+Win64;+x64)+AppleWebKit/537.36+(KHTML,+like+Gecko)+Chrome/52

```

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analyzing Log Files in IIS (Cont'd)



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analyzing Log Files in IIS

Monitoring and analysis of web server logs helps network defenders in determining intrusion attempts or successful intrusions. Web server logs record information about requests made by the users or clients.

Steps to monitor and analyze IIS log files are included below:

- Open the log file in a text editor; the six digits of the log file name represent the day, month, and year when the file was created (e.g., "ex011012.log").
- Trace the header information line that starts with **#Fields:.**; this line is used to determine the corresponding values of each column.
- Identify when the request is created with the date and time; "sitename" and "computername" indicate which server responded to the request.
- Identify who visited the web server using **c-ip** (visitor computer IP address).
- The **cs-method** column contains "post" or "get" requests made by the visitor's browser; **cs-uri-stem** and **cs-uri-query** represent the resource (image/website) requested by the visitor.
- Use the **sc-status** column to find out the capability of the server in responding to requests.
- Use **cs (User-Agent)** to find out which type of browser was used by the visitor.

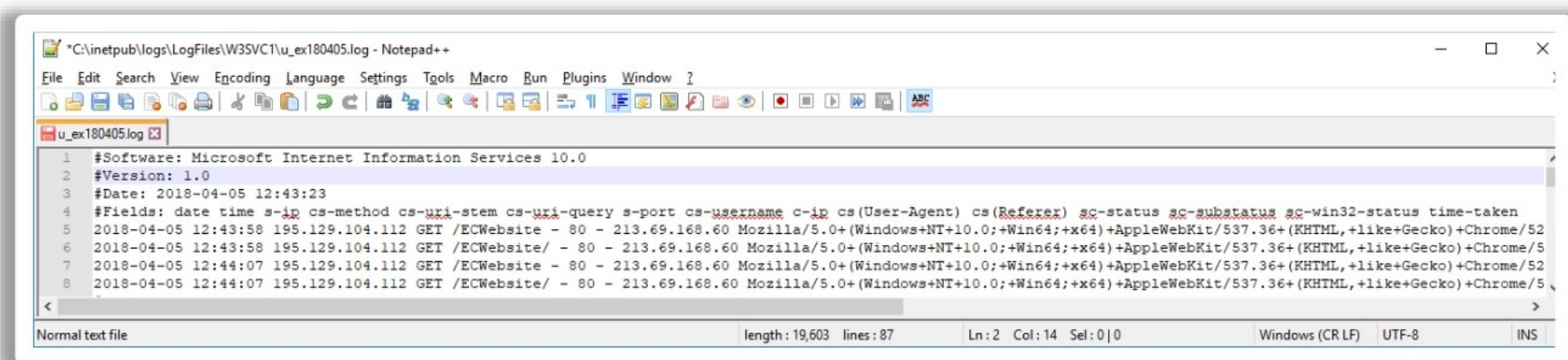


Figure 15.30: IIS Log Files

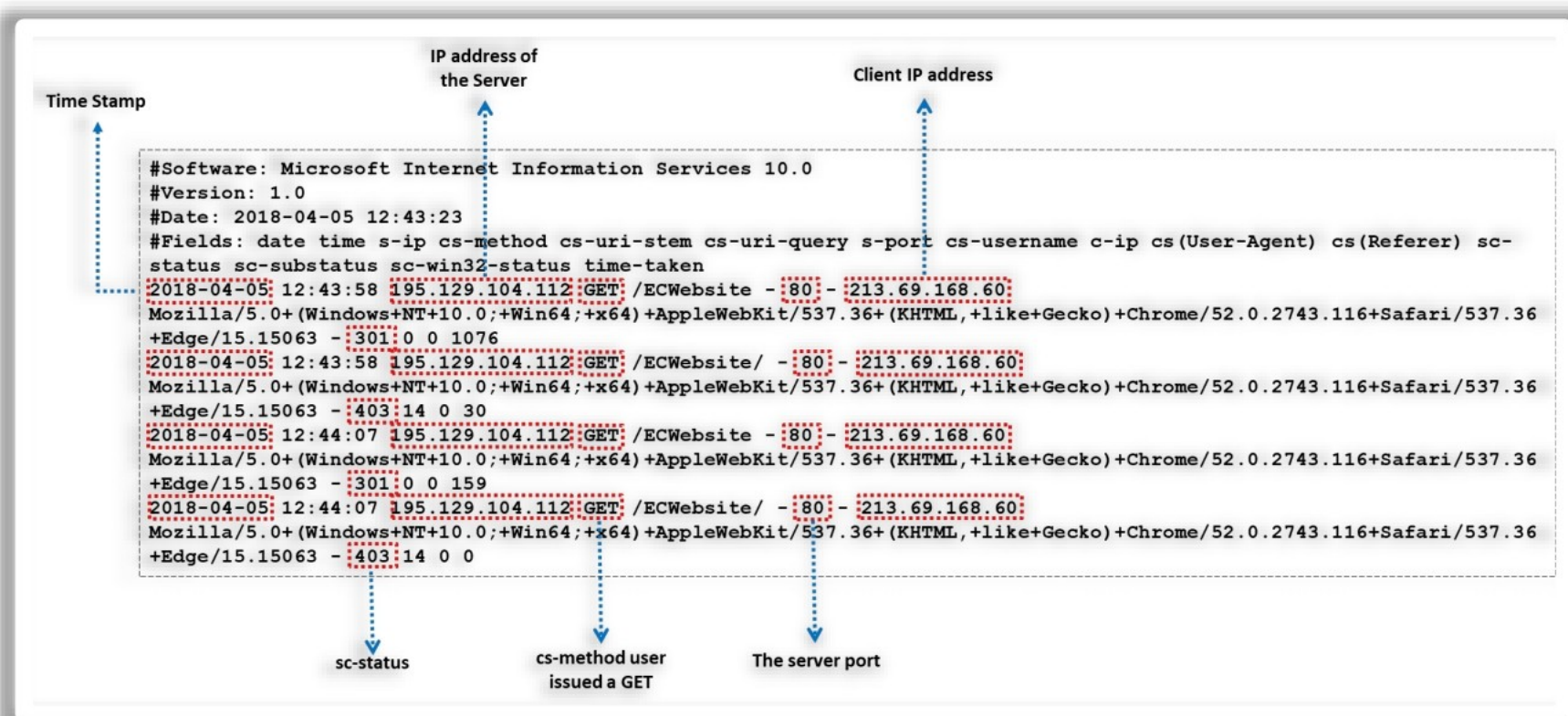
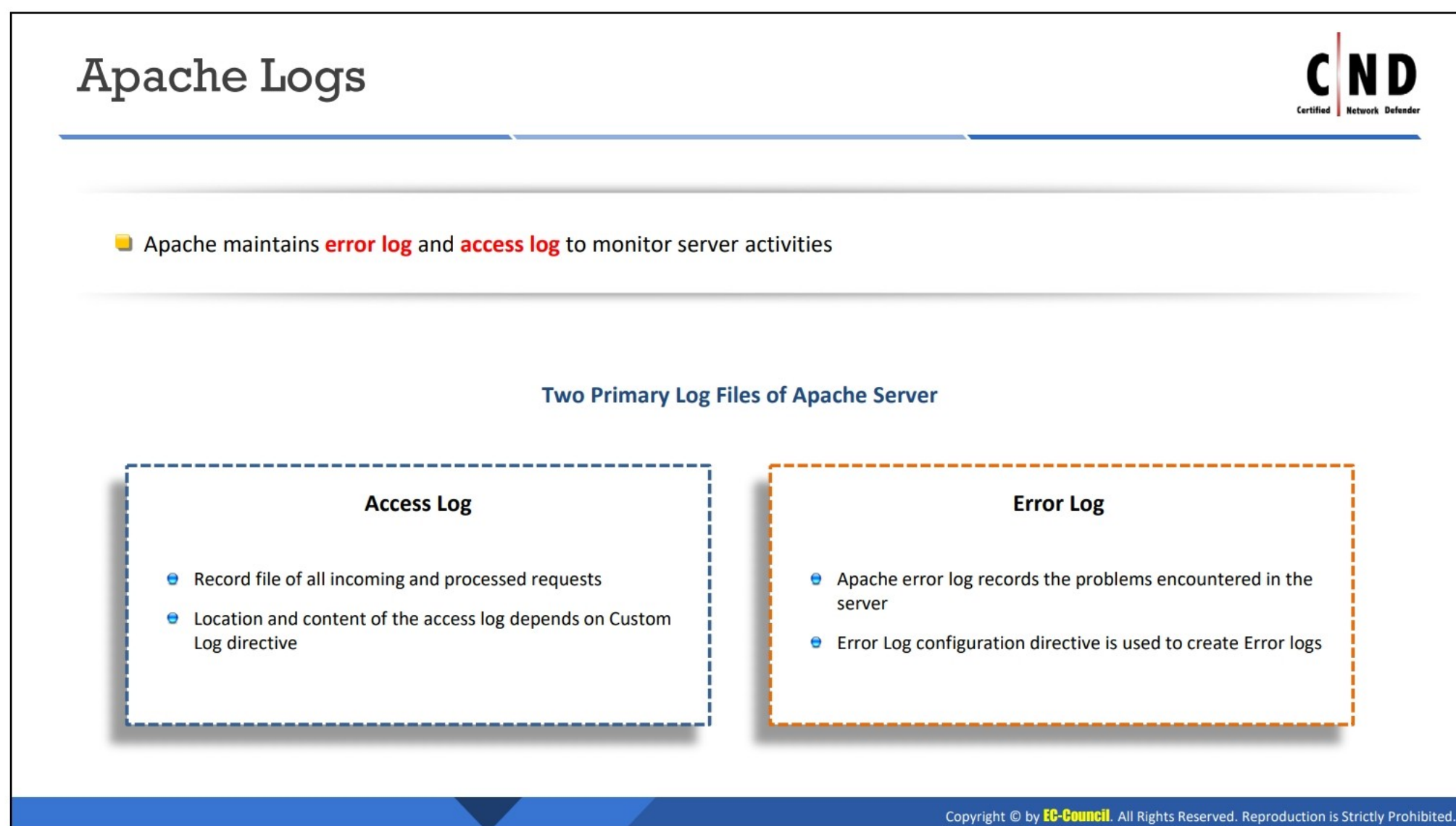


Figure 15.31: Different Parts of IIS Log Files

Log files are created every time a request appears on the server session. The above example shows log file entries in W3C format with properties, date, time, client IP, username, server sitename, server computername, server IP, method, URI stem, URI query, status, time-taken, cookie, and user agent fields.

The time taken field is initialized when the first byte is received by the HTTP server API. This is performed before parsing the request. The time taken field is stopped when the last transmission is completed. The first request to the site takes more time as compared to the remaining requests; this is because the HTTP server API needs to open the log file for logging the first request.



Apache Logs

Apache monitors the usage of the server by extensively tracking the log files. Apache log files are used for logging requests and the actions performed on the server. These files help in managing a web server effectively and determining problems that may have occurred. Apache provides various mechanisms to log everything—from the first request to the final resolution of connection, including errors, alerts, warnings, etc.

Apache server maintains two primary log files: access log and error log.

■ Apache access log

This log records all incoming and processed requests into a log file. Its location and content are managed by the **CustomLog** directive, and its format is highly configurable. The information recorded by access log files helps in analyzing web traffic to the server. The default location of access log files depends upon the distribution:

- In RHEL/Red Hat/CentOS/Fedora Linux, Apache access log files are stored at **/var/log/httpd/access_log**
- In Debian/Ubuntu Linux, Apache access log files are stored at **/var/log/apache2/access.log**
- In FreeBSD, Apache access log files are stored at **/var/log/httpd-access.log**

For example, the following command should be used to view the last 100 lines in the access log.

- If the server is running on RHEL/Red Hat/CentOS/Fedora Linux OS:

```
sudo tail -100 /etc/httpd/logs/access_log
```


- If the server is running on Debian/Ubuntu Linux OS:

```
sudo tail -100 /var/log/apache2/access.log
```

Example of an access log entry:

```
10.185.248.71 - - [09/JAN/2018:19:12:06 +0000] 808840 "GET  
/INVENTORYSERVICE/INVENTORY/PURCHASEITEM?USERID=20253471&ITEMID=2  
3434300 HTTP/1.1" 500 17 "-" "APACHE-HTTPCLIENT/4.2.6 (JAVA 1.5) "
```

■ Apache error log file

This log records the problems encountered in the server. It records both minor problems (such as startup and shutdown messages) and major problems (such as warnings related to specific event and configuration). Its location is configured through the **ErrorLog** directive. In case of any problem, this log file is the first resource to be checked out using **cat**, **grep**, or any other UNIX/Linux command line utilities. This log file provides information regarding problems that have occurred and how to fix them. The default location of the error log file also depends upon the distribution:

- In RHEL/Red Hat/CentOS/Fedora Linux, Apache error log file is stored at `/var/log/httpd/error_log`
- In Debian/Ubuntu Linux, Apache error log file is stored at `/var/log/apache2/error.log`
- In FreeBSD, Apache error log file is stored at `/var/log/httpd-error.log`

For example, use the following command to view the last 100 lines in the error log.

- If the server is running on RHEL/Red Hat/CentOS/Fedora Linux OS:

```
sudo tail -100 /etc/httpd/logs/error_log
```

- If the server is running on Debian/Ubuntu Linux OS:

```
sudo tail -100 /var/log/apache2/error.log
```

Example of an error log entry:

```
[FRI JAN 12 18:04:18 2019] [ERROR] [CLIENT 50.0.134.125] FILE  
DOES NOT EXIST: /VAR/WWW/FAVICON.ICO
```

Log Format

Apache generally uses the common log formats, namely, Apache common log format and Apache combined log format.

- **Apache common log format:** In this log format, basic web log parameters are included. It only displays information that is needed to determine the host and the request. Additionally, information about the agent, cookie string, domain name, referrer, time to serve, etc. is excluded in this format. This format is shown below:

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

The above format string includes percent directives, which direct the server to log a specific piece of information. If the format string includes literal characters, they will

simply be copied into the log output. The character with quotation marks (") can be escaped by placing a backslash before the quotation mark. Special control characters such as \n (for a new line) and \t (for tab) may also be included in this log format string.

The different fields in Apache common log format are shown in the table below:

Fields	Description	Field directive
Client IP Address	Host IP address making the request	%a
Remote Logname	Remote log name. This field is almost always null ("-")	%l
Authenticated Username	Authenticated user's identifier name	%u
Request Date and Time	Date and time at which the request was received by the server (in Common log time format)	%t
Request Line	An HTTP request line that holds the method, request-URI, and protocol ending with <CR><LF>	%r
Status Code	Response code of the HTTP server	%s
Bytes Sent	Data (in bytes) sent from the server to the client	%b, %B
Client Hostname	DNS hostname of the host making the request	%h
Server IP Address	IP address of the host fulfilling the request	%A

Table 15.17: Different Fields in Apache Common Log Format

Example:

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

```
203.93.249.11 - oracleuser [17/Sep/2018:18:45:05 -0700] "GET /files/search/search.jsp?s=driver&a=10 HTTP/1.0" 200 2374
```

The above example is described below.

- **203.93.249.11** (%h) : IP address of the host (remote) making the request
- **-** (%l) : The hyphen represents that specific information, that is, remote log name, is not available
- **Oracleuser** (%u) : Authenticated user's identifier name

- **[17/Sep/2018:18:45:05 -0700] (%t)** : Date and time at which the request was received by the server
- **"GET /files/search/search.jsp?s=driver&a=10 HTTP/1.0" (%r)** : An HTTP request line that holds the method, request-URI, and protocol ending with <CR><LF>
- **200 (%s)** : Response code of the HTTP server. This is very important information. It determines whether the request is responded successfully, redirected, a client error (unauthorized request from the client), or a server error (server is unable to process the request due to some reason)
- **2374 (%b)** : Data (in bytes) sent from the server to the client. If no data is sent from the server to the client, then this value will be denoted as "-." To display 0 in place of "-", use %B instead of %b
- **Apache combined log format:** This log format is similar to common log format but contains two additional fields (referrer and user agent). In other words, it is an extended version of the common log format. Information regarding domain name and transfer time is not provided in this format. This format is shown below:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
```

The additional fields in Apache combined log format are described in the table below:

Fields	Description	Field directive
Referrer	URI of the resource (typically a website) from which the requested URI was obtained	"%{Referer}i"
User Agent	Browser information of the visitor	"%{User-Agent}i"

Table 15.18: Additional Fields in Apache Combined Log Format

Example:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
```

```
203.93.249.11 - oracleuser [17/Sep/2018:18:45:05 -0700]
"GET /files/search/search.jsp?s=driver&a=10 HTTP/1.0" 200 2374
"http://datawarehouse.us.oracle.com/datamining/contents.htm"
"Mozilla/4.7 [en] (WinNT; I)"
```

The two additional fields in the above are described below:

- **"http://datawarehouse.us.oracle.com/datamining/contents.htm" (%{Referer}i)** : It is a "Referer" HTTP request header or URI of the resource (typically a website) from which the requested URI was obtained.
- **"Mozilla/4.7 [en] (WinNT; I)" (%{User-agent}i)** : It is the user-agent HTTP request header or browser information that made the request.

Given below is the name of the parsed fields that are used to parse other log formats:

Fields	Description	Field directive
Filename	Filename of the requested URI	%f
Request Method	HTTP method of the request	%m
Transport Protocol	HTTP protocol version string	%H
Server Port	Port number of the listener fulfilling the request	%p
Server Process ID	Identifier of the process that fulfilled the request	%P
Request Stem	Stem (path) component of the requested URI	%U
Request Query String	Query component of the requested URI	%q
Time to Serve	Time taken to serve the request (in seconds)	%T
Server Name	Server name of the host fulfilling the request	%v, %V
Session Identifier Field	Session identifier as a separate field	
Visitor Identifier Field	Visitor identifier (such as a cookie) as a separate field	
General Purpose Fields 1-10	Users may define (customize) up to ten log fields	

Table 15.19: Name of Parsed Fields

Monitoring and Analysis of Apache Log



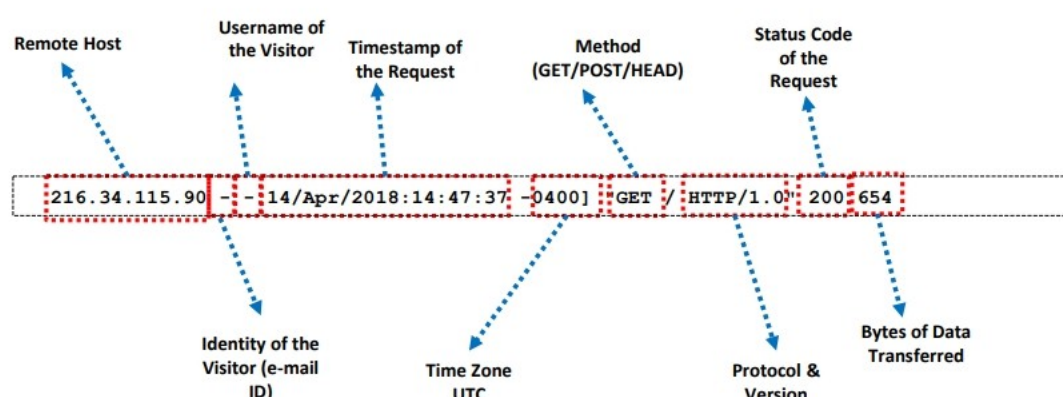
Access Logs

Default apache access log file location in various OSes

FreeBSD: `/var/log/httpd-access.log`

Debian / Ubuntu Linux: `/var/log/apache2/access.log`

RHEL / Red Hat / CentOS / Fedora Linux: `/var/log/httpd/access_log`

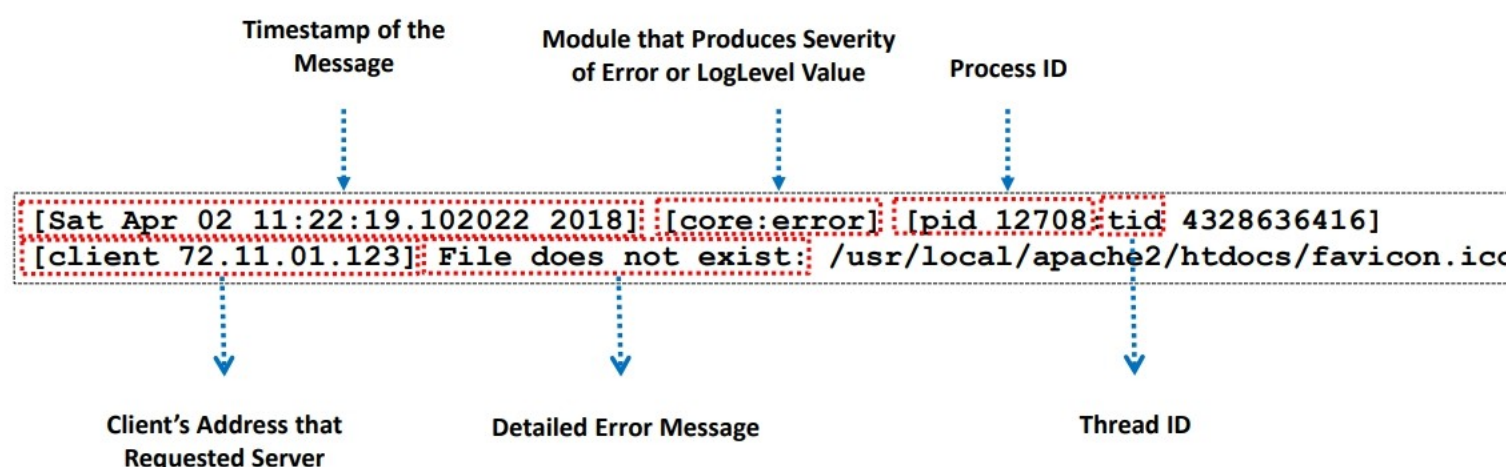


Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Apache Log (Cont'd)



Error Logs



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Monitoring and Analysis of Apache Log

The Apache access and error logs provide actionable insights regarding potential server configuration and web application problems. However, a concern with them is that important information is concealed inside a large number of log messages. Therefore, the goal of Apache log analysis is to extract only the important information to gain an understanding about the issues and how to respond to them before they affect the users. However, monitoring of Apache access and error logs is required before analyzing them.

Monitoring Apache Access Log

- To monitor Apache access log file, navigate to one of the following two directives based on the OS:

`/var/log/httpd/access_log`, or

`/var/log/apache2/access.log`

- If the Apache access log file is unreachable at the given path, then it may be due to a custom configuration in the Apache config file. In this case, open the Apache configuration file `httpd.conf` to find the location of the access log file.

Monitoring Apache Error Log

- To monitor Apache access log file, navigate to one of the following two directives based on the OS:

`/var/log/httpd/error_log`, or

`/var/log/apache2/error.log`

- Apache does not allow use of a custom error log format.



Monitoring and analyzing log files of different devices locally can be a difficult task. Centralized logging helps you to simplify the process.

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

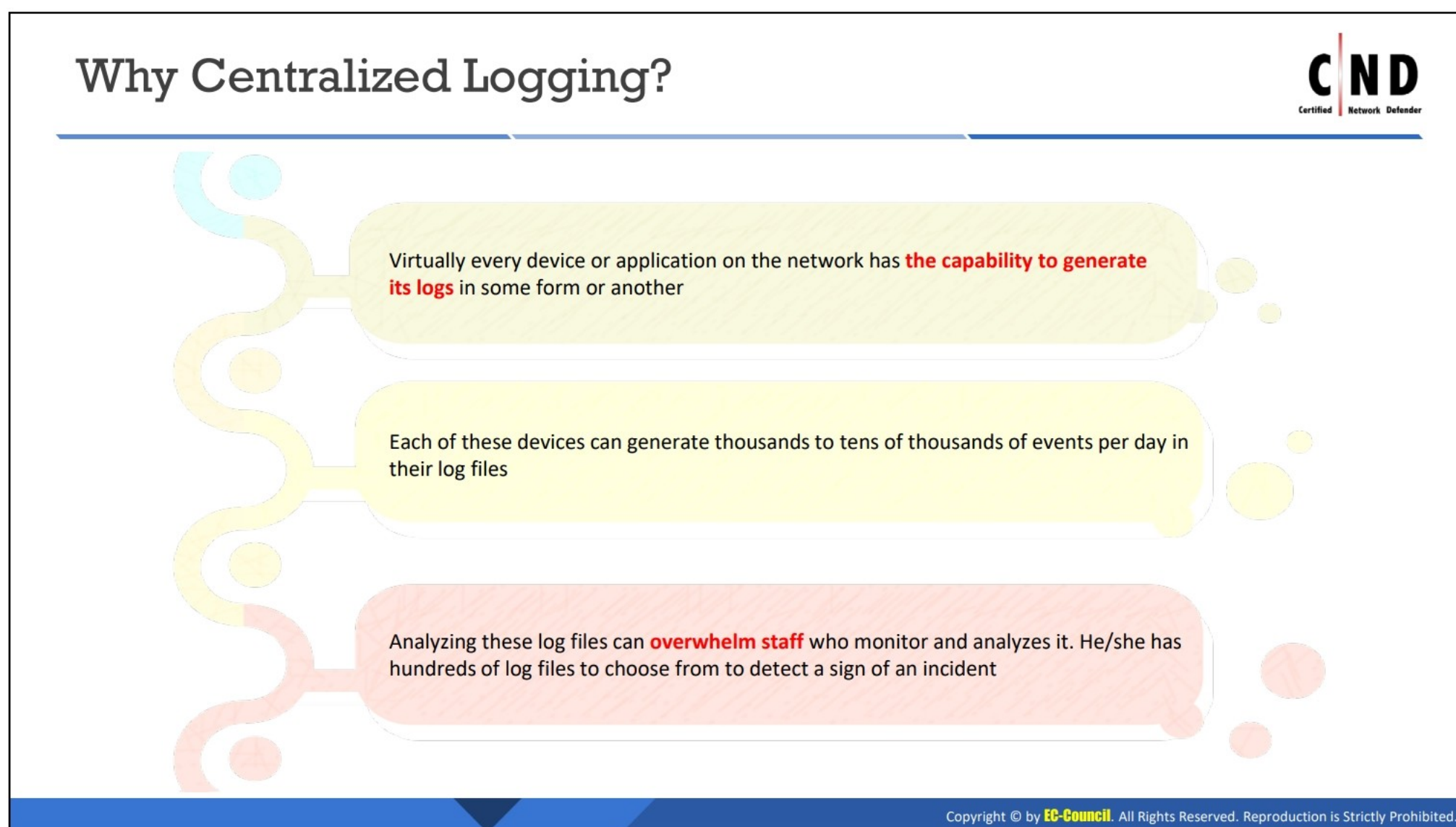


LO#08: Discuss centralized log monitoring and analysis

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

LO#08: Centralized Log Monitoring and Analysis

This section describes the importance of centralized log monitoring and the process to be followed for centralized log monitoring and analysis.




Why Centralized Logging?

An organization that runs on a number of devices and uses several applications generates thousands to tens of thousands of events every day in its log files. The events may include user session activity, all access attempts (including failures), the connection of external devices to the organization's network, and other events. Analyzing all these log files would be a difficult task for the staff who monitors and analyzes them. Centralized logging can help maintain the records of all the log files in one place, where they can be accessed, monitored, or scheduled for alerts against threats. Centralized logging helps provide deep insights about the network events such as what kind of event it is, when and why it is generated, etc. without any hassle. Even if the staff is not monitoring the logs, the centralized logging application can review the logs, pick out suspected events, and alert the network defender through various means. This helps the management respond faster and take the required steps.

Besides this, there are several reasons for adopting centralized logging over local logging:

- With centralized logging, log analysis outside the production environment is possible.
- Centralized logging makes it easy to search for a single transaction that has been processed across multiple application servers.
- Centralized logging is a storage efficient technique that helps in saving costs and reducing disk I/O on application servers.

Centralized Logging



- In centralized logging, logs from different devices and applications on the network are collected to the one **central location**
- This helps staff to clearly and quickly monitor, analyze, and review the logs for any anomalies

Centralized Log Management Capabilities:

- Stores logs from different sources to one central location
- Easily access the important data from logs files
- Generates alerts based on metrics defined in the log
- Quickly share the dashboard and log information with others

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Centralized Logging

Centralized logging or centralized log management (CLM) is a logging solution that can collect all the log files generated by different devices and applications on the network and push it to one central and accessible location. It not only collects log information but also supports aggregating, monitoring, analyzing, and reviewing logs for any anomalies.

CLM typically provides the following capabilities:

- It allows a scan of the entire network in one location and obtain a 360-degree view of the activities that are going on in the network.
- It generates alerts based on the metrics defined on the log.
- It allows sharing of the dashboard and log information with others.
- It allows access to the important data from logs files.
- It applies retention policies so that a specific log can be available at the right time.
- It provides a large amount of space for storing and taking a backup of historical data.
- It provides a deep understanding of the events and helps the staff answer questions such as who, what, when, and where an event occurred.
- It improves the security of the network.
- It allows access to data in seconds instead of hours, weeks, or even days.

CLM processes the log files by collecting, transporting, storing, and analyzing them and then decides whether to send an alert or not, based on the configured rules. The alerts can be sent through different means such as emails, help desk tickets, or in any other way the user prefers.

Centralized Logging Infrastructure



Log management architecture generally consists of **three** different tiers such as:

1

Log Generator: It consists of the **host** that generates the log data

2

Log Analysis and Storage:

Collection Server

- It consists of one or more log servers that receives log data
- The log servers which can collect the log data also called as log collectors or aggregators
- It uses different protocols to collect logs like syslog, SNMP, etc.

Storage Server

- It stores the collected log data on log server or separate database server like Oracle, MS SQL, etc.

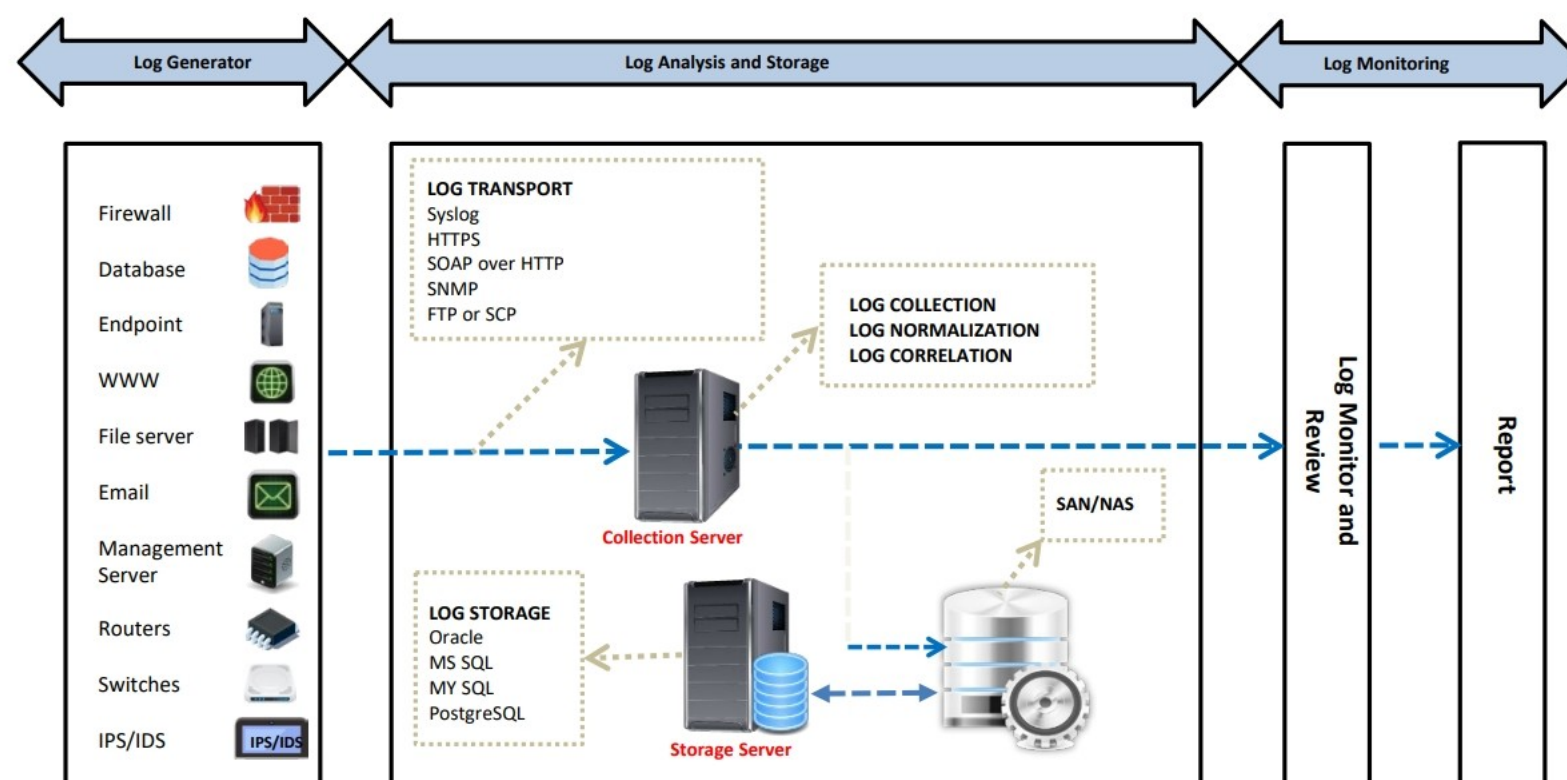
3

Log Monitoring:

- It is used to **monitor** and **review** the log data
- It can also be used to generate reports

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Centralized Logging Infrastructure (Cont'd)



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

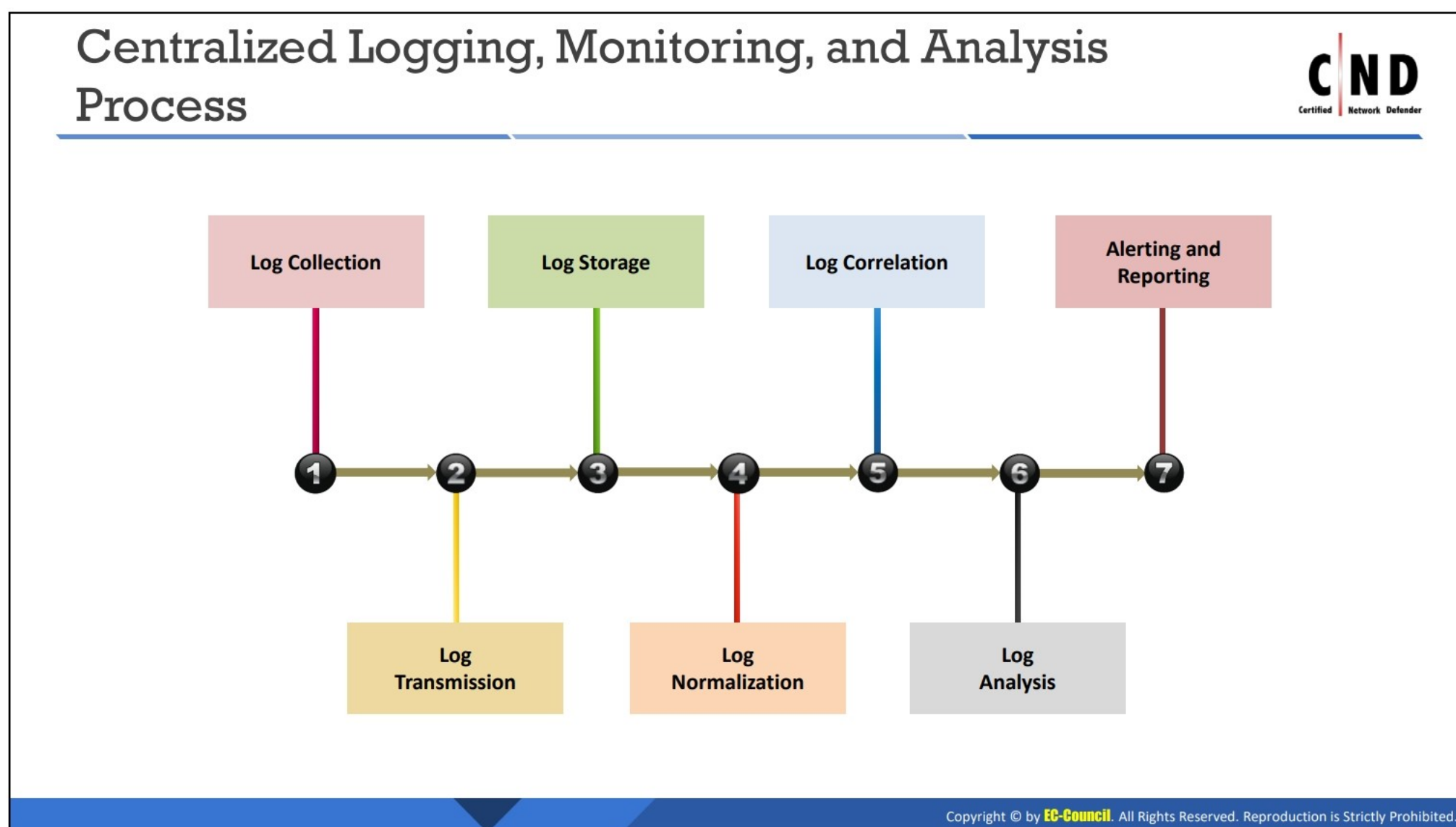
Centralized Logging Infrastructure

A log management infrastructure is a combination of hardware, software, networks, and media that generate, transport, store, analyze, and report log data. More than one log management infrastructures can exist in an organization. Log management architecture generally consists of three different tiers.

- **Log generation/generator:** This tier consists of the host that produces the log messages. Some hosts use logging client applications or services to transfer their logs to log servers, while others prefer to do the same through other means. In some cases, the generator of the data may be a router, switch or a firewall, application, database, etc.
- **Log analysis and storage:** This tier consists of one or more log servers that collect log data from the hosts. The log data can be sent either in real-time or in batches based on the schedule to the log server. The log servers that are able to collect log data also called collection servers or aggregators. They use different protocols to collect the logs such as syslog, SNMP, etc. Log messages can be stored either in collection servers or on separate database servers.
- **Log monitoring:** The third tier consists of consoles that monitor and review the log data as well as the outputs of log analysis. These consoles are used to produce reports. They may also manage log servers and clients. One can also limit console user privileges to required functions and data sources.

The second tier, that is, log analysis and storage, can differ in complexity and structure. The simplest configuration among all is that a log server is managing all log analysis and storage functions. Some complex configurations are described below:

- Numerous log servers where each one performing a specific operation such as log collection, log analysis, long-term storage, etc.
- Numerous log servers and each one is analyzing and storing logs for a few log generators
- Two levels of log servers, where the first level of distributed log servers transfer logs to second-level centralized log servers



Centralized Logging, Monitoring, and Analysis Process

In centralized logging, logging, monitoring, and analysis of logs are performed through a series of steps.

The typical series of steps involved in the process of centralized logging, monitoring, and analysis is as shown in the below figure:

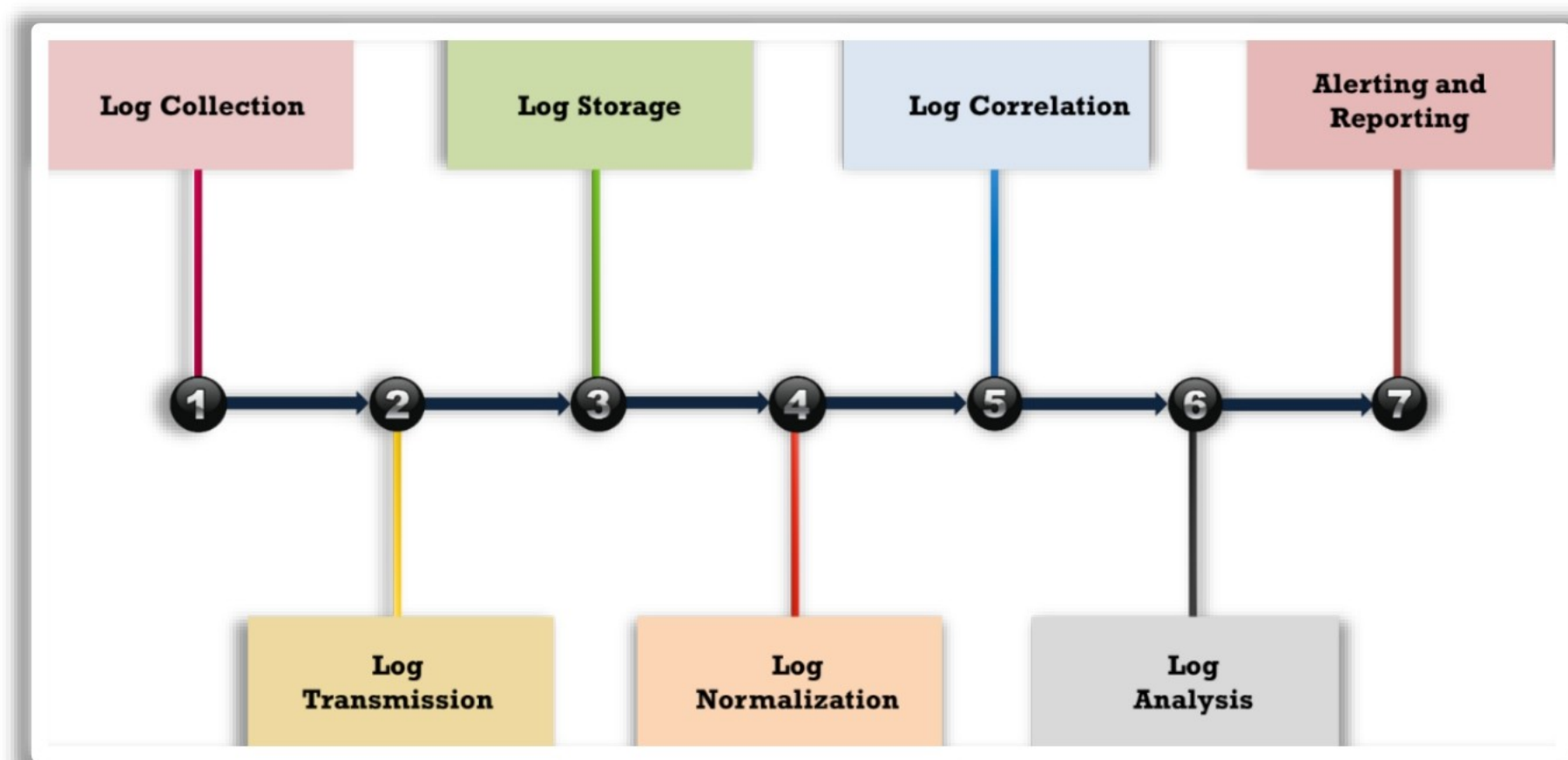
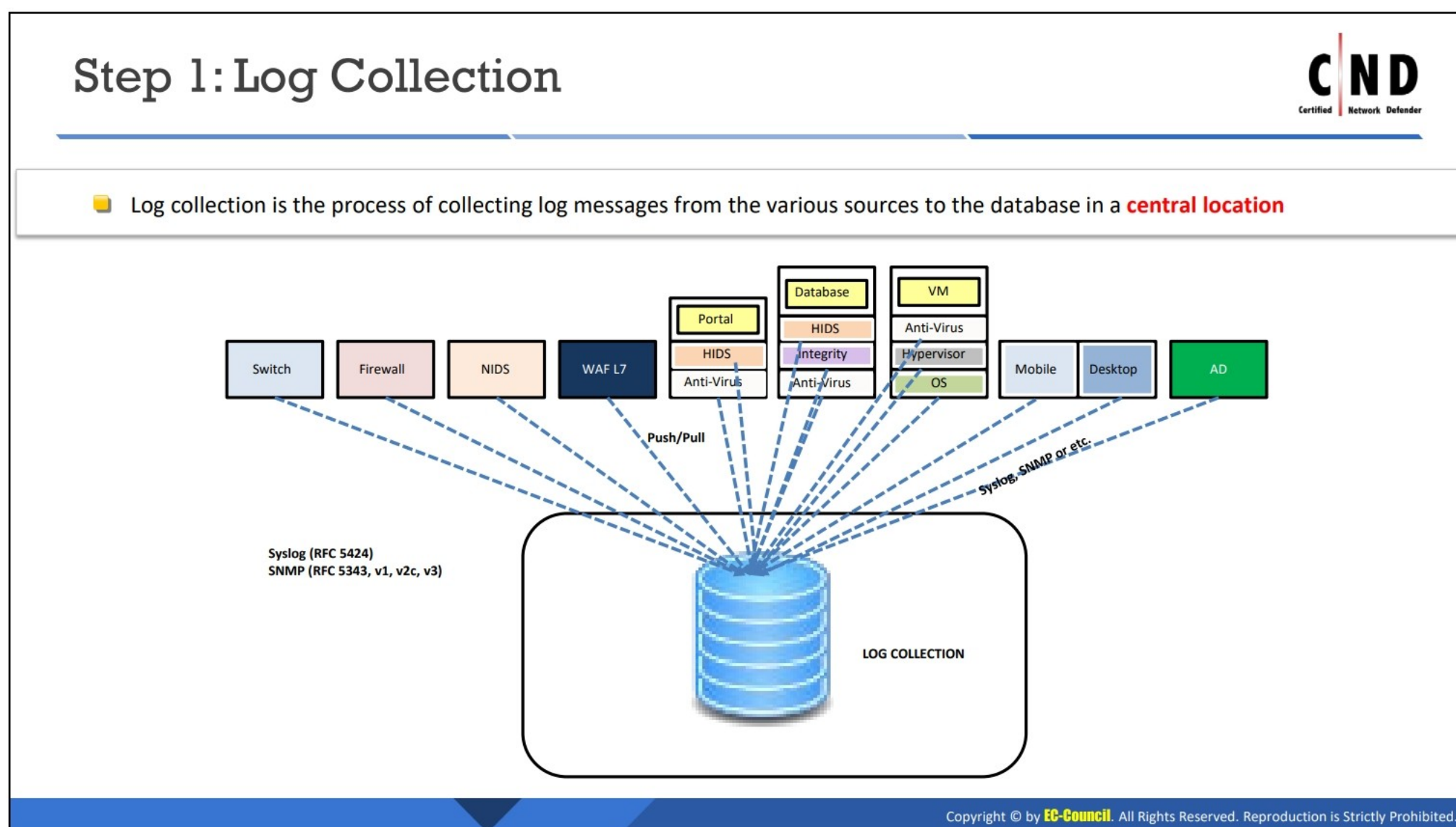


Figure 15.32: Steps Involved in Centralized Logging, Monitoring, and Analysis

- **Log collection:** Log collection is the process of collecting log messages from the various sources to the database in a central location.

- **Log transmission:** To store the logs in a centralized location, they are transmitted through different transport mechanisms such as syslog UDP, syslog TCP, encrypted syslog, etc.
- **Log storage:** All the log files collected from various devices are stored in central repository/databases. Stored databases can be retrieved in structured way when needed.
- **Log normalization:** Log normalization is the process of accepting logs from heterogeneous sources with different formats and converting them into a common format.
- **Log correlation:** Log correlation is the process of matching a series of normalized log data to determine a set of related events based on a certain set of rules.
- **Log analysis:** Log analysis is the process of identifying the patterns and anomalies in the correlated log data that signifies any intrusion attempt or policy violation activity.
- **Alerting and reporting:** An alerting system generates alerts and sends a report to the user if any suspicious event is observed in the logs or calculated matrices.

Note: A detailed explanation of each phase of this process is described in the upcoming slides.



Step 1: Log Collection

Various security systems such as antimalware tools, proxies, firewall, authentication servers, routers, switches, etc. generate log messages. Even OS and web applications generate a wide range of log messages. These log messages cover a wide range of information such as user IDs, system activities, timestamps, successful or unsuccessful access attempts, configuration changes, network address and protocols, file access activities, etc. The process of collecting these log messages from various log sources and collating them to a central database is known as log collection. This operation is performed by a log collector. The transmission between the log collector and a central log server is encrypted to avoid eavesdropping.

Log collection at a central location provides many advantages, a few of which are discussed below.

- **Redundancy:** Log messages are kept in more than one location.
- **Store and forward:** If the log collector loses connection to the central log server while forwarding the log messages to it, it will store those log messages and forward them when the connection is reestablished. This prevents possible data loss.
- **Authentication:** The log collector not only verifies the sender as a trusted source but also the server to whom it is sending log messages.
- **Privacy:** The transmission between the log collector and log server is kept private through data encryption.

Step 2: Log Transmission



- Logs are transmitted to a central location using various **log transport mechanisms**

- Typical log transport mechanisms are:

- Syslog UDP
- Syslog TCP
- Encrypted Syslog
- HTTP
- HTTPS
- SOAP over HTTP
- SNMP
- File transfer protocols such as FTP or SCP

- An efficient log **transport mechanism** should:

- Maintain integrity, availability, and confidentiality of log data
- Maintain log format and meaning
- Represent all the events correctly with perfect timings and event sequence

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Step 2: Log Transmission

A process of moving log messages to a central location is known as log transport. An efficient and reliable mechanism should be used to transmit log messages. It should be able to maintain integrity, availability, and confidentiality of log data, log format, and meaning of data as well as represent all the events correctly with perfect timings and event sequence.

Log data can be transmitted through log transport protocols such as syslog UDP, syslog TCP, etc. Some of them are described below.

- **Syslog UDP:** Syslog User Datagram Protocol is faster at transferring log data compared to TCP. This is because it does not wait for the server to confirm whether information is received or not. Despite this weakness, this is one of the most popular log transport mechanisms.
- **Syslog TCP:** Syslog Transmission Control Protocol first establishes a connection to the server and then transmits the data to it. Once the data is transmitted, it waits for the server to confirm receipt of information through an acknowledgment message. It also has flow control capabilities.
- **Encrypted syslog:** Syslog is a clear-text protocol. Encrypted syslog was introduced to make sure the transmitted log data was encrypted; this makes transport of logs over TCP/UDP secure.
- **HTTP/HTTPS:** HTTP/HTTPS can be used in transferring log data between devices and can also be used to send and receive files based on TCP/IP protocols.
- **SOAP over HTTP:** The Simple Object Access Protocol is another messaging protocol, and it is also used for transmitting syslog. This process is done over an HTTP payload as HTTP is an application protocol.

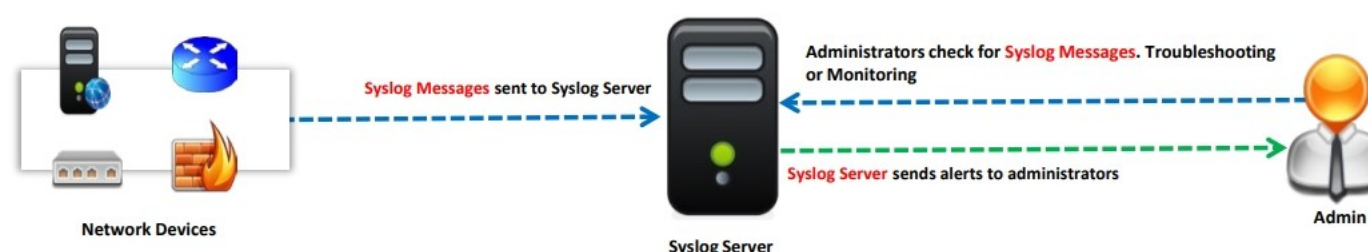
Example: Syslog Log Transport Mechanism



- Syslog is a **data logging service** that enables network devices such as routers, switches, firewalls, printers, web-servers, etc. to send and store logging of events and information on a logging server
- Logging server is a dedicated server called **syslog server** and events send are called **syslog messages**
- Syslog stores consolidate logs from multiple devices into a single location

Components of syslog:

- Syslog listener
- Database
- Management and filtering software



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

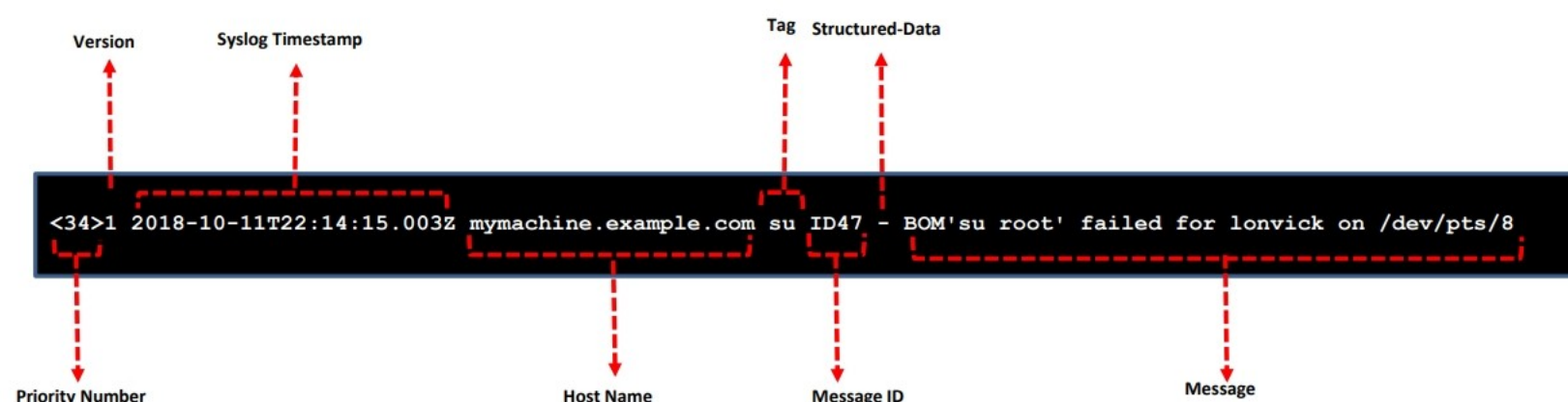
Example: Syslog Log Transport Mechanism (Cont'd)



Syslog Message Format:

```
<priority>VERSION TIMESTAMP HOSTNAME TAG MESSAGEID STRUCTURED-DATA MSG
```

Syslog Message Example:



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Example: Syslog Log Transport Mechanism

System Logging Protocol (syslog) is a standard for data logging service; network devices use to forward log messages to a server across an IP network. A syslog server is a central repository that stores logs from various network devices. It consolidates logs from multiple devices (switches, firewall, routers, and others) into a single location. Events logged in IDS and IPS are also sent to the syslog server using internet protocols such as TCP, UDP, HTTP, HTTPS, SNMP, etc.

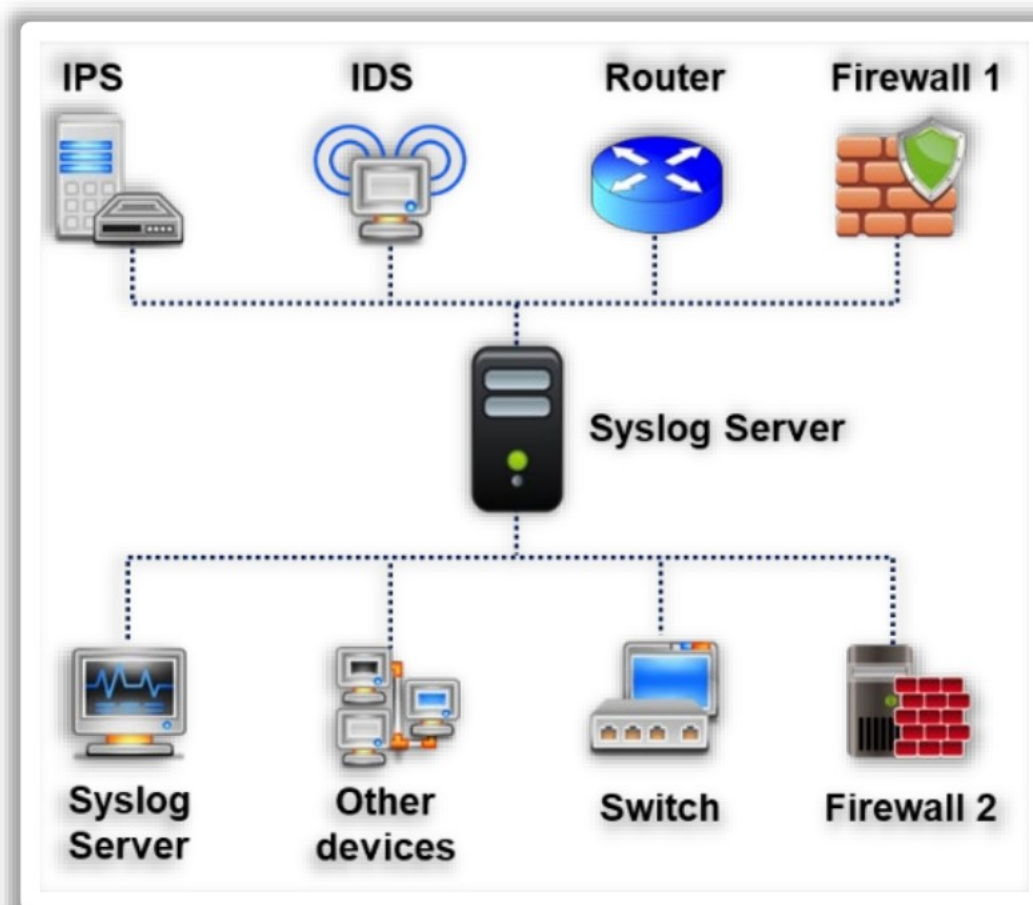


Figure 15.33: Syslog Server

A syslog server provides centralized log management for storing messages received from multiple network devices. This simplifies the process of monitoring and analyzing the security posture of all network devices. It generates alerts when any suspicious activity is generated or when prenotified events occur.

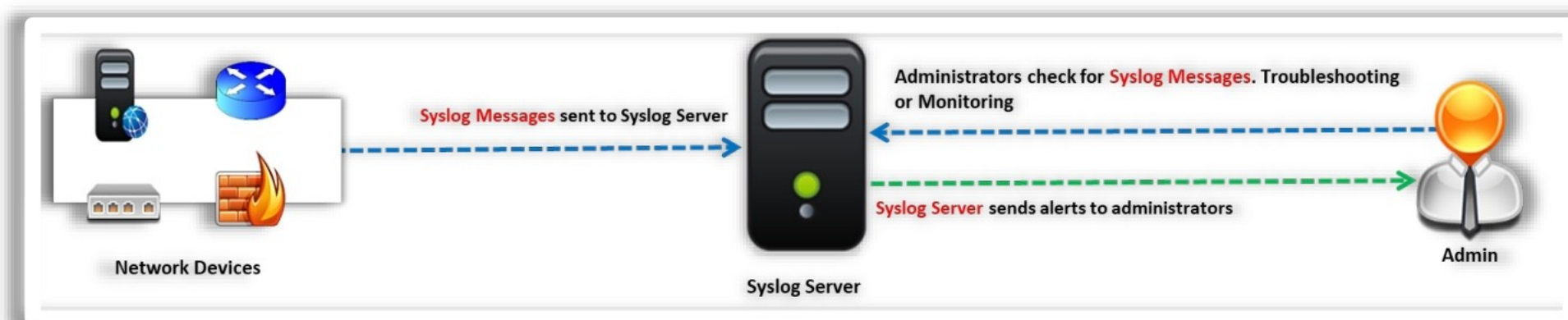


Figure 15.34: Working of the Syslog Server

Components of Syslog Server

A syslog server can perform different tasks with the gathered information such as giving access to the network defenders quickly, monitoring the events and analyzing them, and sending alerts for suspicious events, etc. There are separate components in a syslog server for each of these tasks. They are described below.

- **Syslog listener:** A syslog listener gathers the log messages that are sent by the devices over the network using UDP port, which is the standard syslog port but does not provide acknowledgment message when receiving the log messages. Therefore, a TCP port can be used for this purpose. Syslog also listens to the data sent over different ports such as HTTP and HTTPS as it works on a layered architecture.
- **Database:** Syslog database is the place where all the log messages that are being sent to the syslog server are stored. The information may include system activities, unsuccessful

attempted events, devices that are connected to the network, etc. This information is indexed and can be retrieved when necessary.

- **Management and filtering software:** Log messages contain a wide range of information, and to extract important information from it will be a time-consuming task. Therefore, the syslog server takes the help of management and filtering software to perform this task. This software filters the messages and provides notification to network defender about detected errors. For example, it filters and displays all critical log messages related to the firewall. It also uses negative filter rules to avoid notifications regarding certain types of entries.

Different Roles in Syslog

- **Originator:** The entity that generates the syslog message is known as an originator. In some cases, the originator may be a router, switch, or a firewall. The data generated in the originator could be regarding the activities in an OS, connection of external devices to the network, installation of third-party applications in a system, etc.
- **Relay:** This is an entity that receives the messages from the originator and forwards them either to syslog relay or syslog collector in the network. There may be multiple syslog relays in a network. For example, if a company has a branch office at another location and the log files of that branch office are also to be stored at the centralized location, it may not be done in one direct step. In such a case, the log data is sent to a central machine through syslog relay.
- **Collector:** This is the entity that receives the information of the event in the server in a syslog format. The information generated by the originator is collected in the syslog collector. The format in which the information is generated may be of any kind and is specified by the standard server. The syslog collector is often referred to as syslog server.

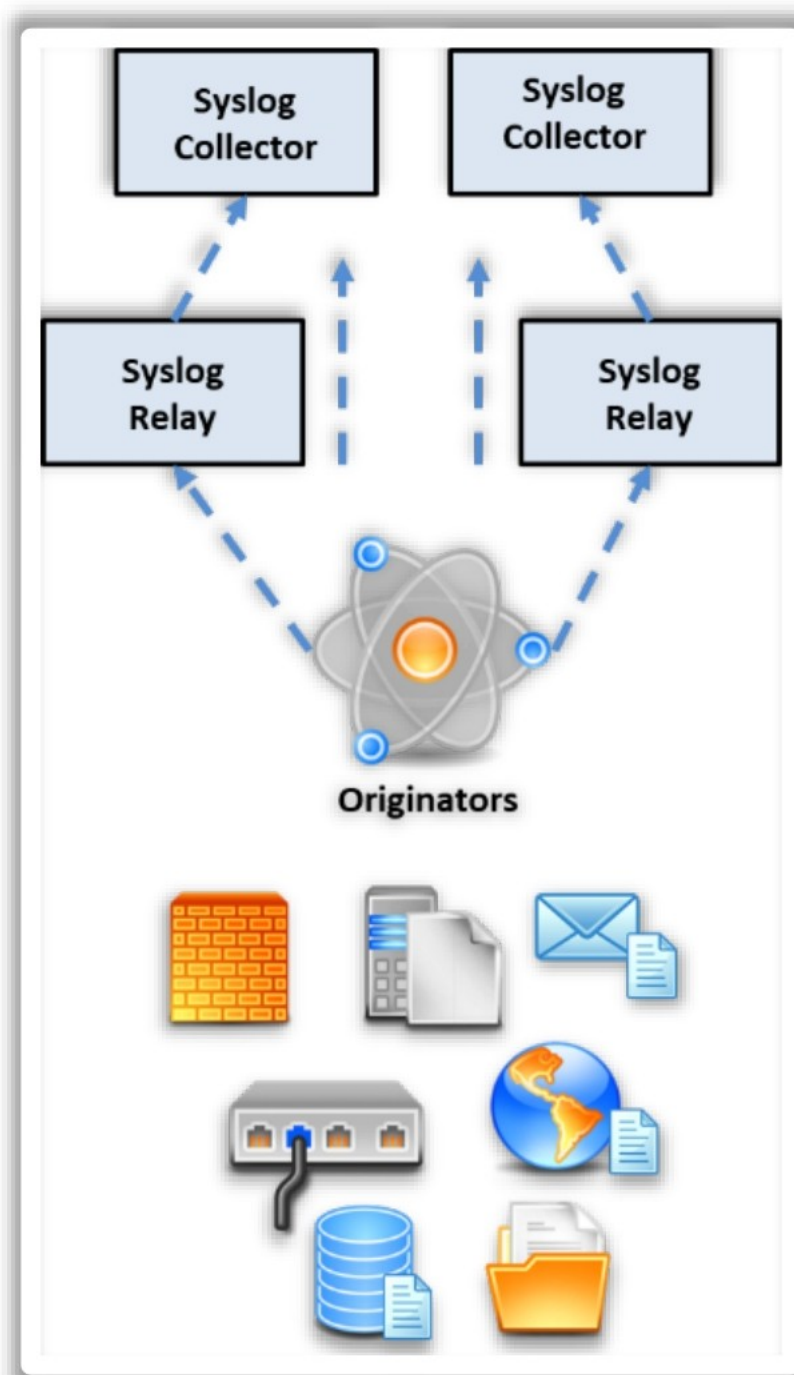


Figure 15.35: Different Roles in Syslog

Different Layers of Syslog

According to the syslog standard, there are three different layers—syslog transport layer, syslog application layer, and syslog content layer. Syslog transport layer transmits the log data/messages over the network; syslog application layer interprets, routes, and stores the log messages; and syslog content layer includes the actual information kept within the log messages.

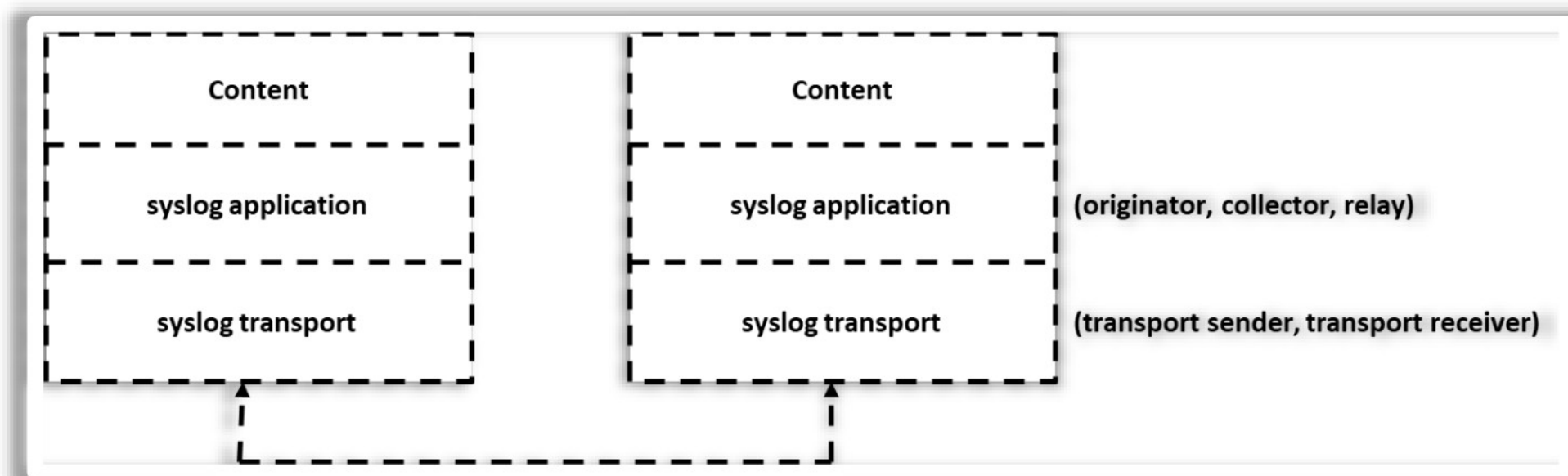


Figure 15.36: Different Layers of Syslog

- **Syslog content:** Syslog content layer includes those devices that generate log messages that are to be sent to the syslog server. It contains the actual message that is to be sent, including audit record, events record, etc.

- **Syslog application:** Syslog application layer manages generation, interpretation, routing, and storage of syslog messages. This layer consists of the originator (which generates the messages), collector (which collects the messages sent by the originator), and relay (which also collects the information from the originator and forwards it to syslog collector or another syslog relay).
- **Syslog transport:** In the syslog transport layer, the log message is transmitted over the network and maps to the actual transports. It consists of a transport sender (which sends the log messages to a specific transport protocol) and a transport receiver (which receives the log messages from the specific transport protocol). In the transport layer, the "framing" technique is used. In this technique, assembling of all log messages at the source side and disassembling of those log messages at the receiver side is done. Each message in this process is described as a frame.

Syslog Message Format


The log events and information that are sent are called syslog messages. Syslog messages include important information that provides deep insights about when, where, and why a specific log message was generated. It follows a standard format for sharing messages among the application.

Message formats are of different types:

- RFC3164, which is an old format introduced in 2001. This was the standard BSD format.
- RFC5424, which is the new standard format that is currently in use; it was introduced in 2009 to overcome the problems of RFC3164.

The standard message format of RFC5424 consists of three components, as described below.

- **Header:** This field comprises subfields for priority, version, timestamp, hostname, application, process ID, and message ID.
- **Structured data:** It contains data blocks in the "key=value" format.
- **Message:** It should be UTF-8 encoded and it contains a description of the event generated.



The diagram shows a single line of text representing the Syslog message format: <priority>VERSION TIMESTAMP HOSTNAME TAG MESSAGEID STRUCTURED-DATA MSG. This text is enclosed in a dark rectangular box with a white border, which is itself centered within a larger white rectangular area.

```
<priority>VERSION TIMESTAMP HOSTNAME TAG MESSAGEID STRUCTURED-DATA MSG
```

Figure 15.37: Syslog message format

Example of a syslog message:

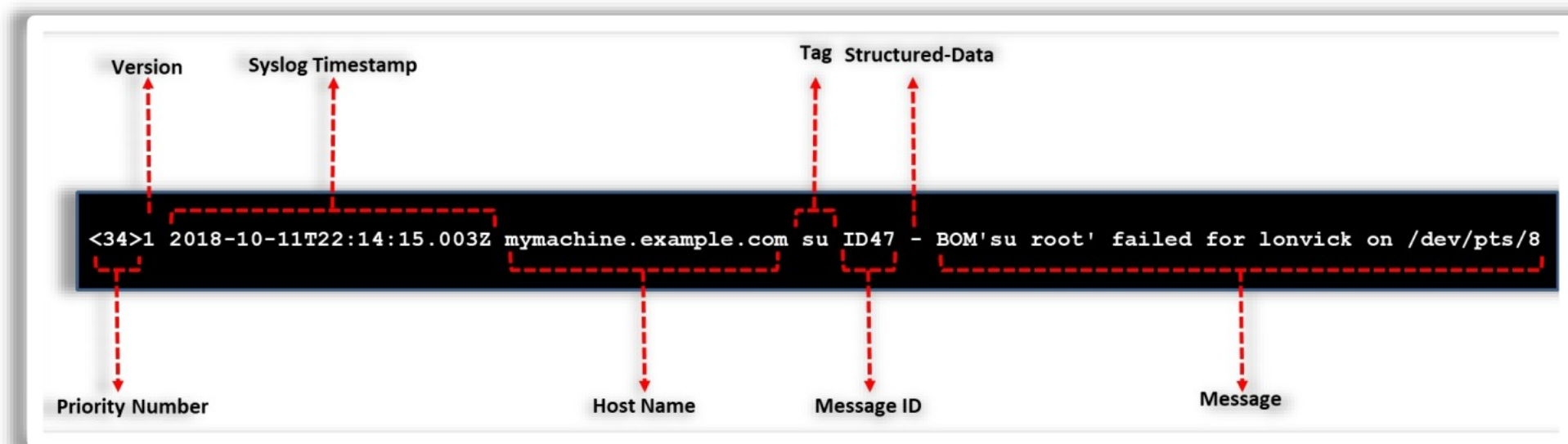


Figure 15.38: Example of a Syslog Message

Calculation of Priority Value

Priority value, which is also known as PRI, is used to represent the facility and severity of the message. Facility provides information about the sender of the message, and severity represents the importance of the message.

The priority value is present at the beginning of the message and is enclosed in "<" and ">". The priority value exists between 0 and 191 and is calculated by the formula:

$$\text{Priority value} = (\text{facility value} \times 8) + \text{severity value}$$

Each of these values is assigned based on the categorization described in the tables below. If the PRI value of an event is low, then the priority of that event is high.

Numerical code	Facility
0	Kernel messages
1	User-level messages
2	Mail system
3	System daemons
4	Security/authorization messages
5	Messages generated by syslogd
6	Line printer subsystem
7	Network news subsystem
8	UNIX-to-UNIX copy (UUCP) subsystem
9	Clock daemon
10	Security/authorization messages
11	TP daemon
12	NTP subsystem

13	Log audit
14	Log alert
15	Clock daemon
16–23	Locally used facilities (local0–local7)

Table 15.20: Values of Facilities

Level	Description
Emerg	Emergency system is unusable
Alert	Action must be taken immediately
Crit	Critical conditions
Error	Error conditions
Warn	Warning conditions
Notice	Normal but significant condition
Info	Informational
Debug	Debug-level messages

Table 15.21: Values of Severity

Header

According to the standard message format, the header must contain 7-bit ASCII character set in the 8-bit field. It is the metadata of the event logs. It consists of identifying information of a syslog message such as a timestamp, hostname, or an IP address.

The header part of the syslog message consists of a timestamp and the hostname or IP address. If the system in which the message is generated does not contain a hostname, then the message header will contain IP address in its part. The timestamp in the header is a combination of date and time at which the message was generated. The format of the timestamp is in the local time, in the Mmm dd hh:mm:ss format.

Message


The MSG part contains a TAG field (which represents the name of the program that has generated the message) and a CONTENT field (which includes the details of the message itself). The TAG field of the message is further divided into three fields.









- **APP-NAME:** The APP-NAME is used to identify the originator of the message. If the system cannot provide the information, then NIL value is assigned to the message. No value is assigned if the information is not available for that device.
- **PROCID:** The PROCID is used to identify the process name or process ID. When a process is not available, then PROCID is set to NIL value. A change in PROCID indicates that there

is a discontinuity in syslog reporting. However, it is not reliable for a restarted process as such a process is assigned the previous process ID.

- **MSGID:** The MSGID is used to identify the type of message sent. There are different types of messages that may be coming out or going in. If the messages are coming out from a UDP port, then they are designated by MSGID as "UDPOUT." However, if they are going in, then they use "UDPIN" as their MSGID. Messages with the same MSGID should reflect events of the same semantics. MSGID is a string, and its main purpose is to filter the messages that are passing over the relay and collector.

Syslog Tools



 Kiwi Syslog Server https://www.kiwisyslog.com	 SNMPSoft Sys-log Watcher https://www.netadmintools.com
 Splunk Light https://www.splunk.com	 Visual Syslog Server https://www.github.com
 WhatsUp Gold https://www.whatsupgold.com	 Fastvue Syslog https://www.fastvue.co
 Syslog-NG https://www.syslog-ng.com	 NxLog https://nxlog.co

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Syslog Tools

The following are few syslog server tools.

Kiwi Syslog Server

Source: www.solarwinds.com

Kiwi Syslog Server provides centralized and simplified log message management across network devices and servers. It allows managing syslog messages, SNMP traps, and Windows event logs.

SNMPSoft Syslog Watcher

Source: www.ezfive.com

SNMPSoft Syslog Watcher for Windows that collects, parses, stores, analyzes, and explains syslog messages to professional network administrators and helps improve the stability and reliability of the network.

Splunk Light

Source: www.splunk.com

Splunk Light automates log search and analysis, as well as server and network monitoring. It centrally collects and indexes all the log data including syslogs, event, web, and IIS logs regardless of format or location. It builds dashboards around security compliance, clickstream data, and website transaction failures. Splunk Light also maximizes uptime of network, operational, and e-commerce servers through real-time alerts.

Visual Syslog Server

Source: www.github.com

Visual Syslog Server for Windows is useful when setting up routers and systems based on Unix/Linux. It provides live messages view, switches to a new received message, color highlighting, message filtering, customizable notifications and actions, etc.

WhatsUp Gold Syslog Server

Source: www.whatsupgold.com

WhatsUp Gold Syslog Server collects and stores syslog messages, thereby providing a reliable central repository for log data.

Fastvue Syslog

Source: www.fastvue.co

Fastvue Syslog can detect incoming syslog data and automatically log the messages to organized text files. It automatically zips logs older than 30 days (configurable) and moves them to an archive folder, reducing disk space requirements. It provides automatic SHA256 hash file for each log for validation. It allows forwarding syslog messages to other syslog servers.

syslog-ng

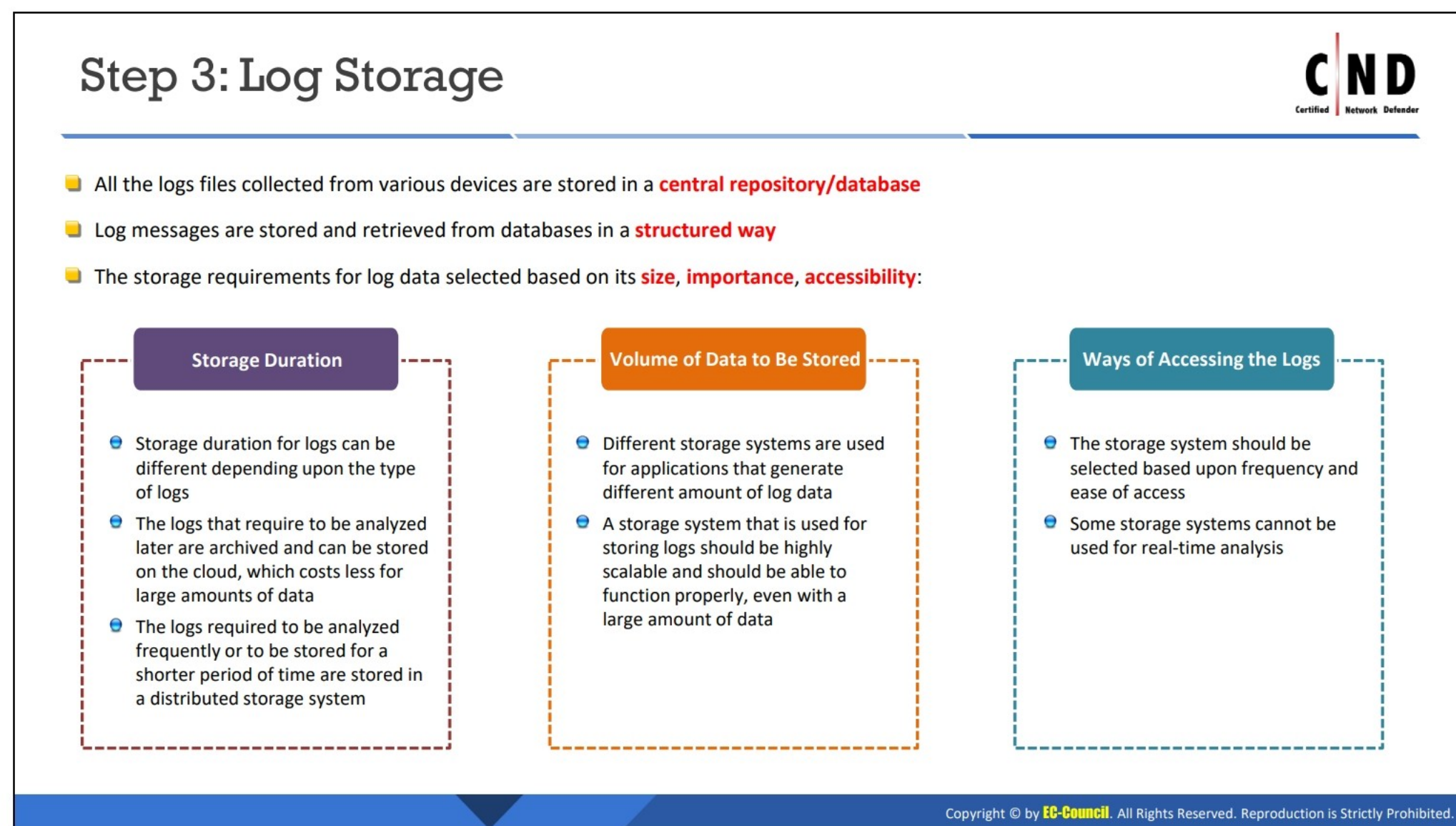
Source: www.syslog-ng.com

The syslog tool syslog-ng collects logs from any source, processes them in real time, and delivers them to a wide variety of destinations. It allows to flexibly collect, parse, classify, rewrite, and correlate logs from across the infrastructure and store or route them to log analysis tools.

NxLog

Source: <https://nxlog.co>

NxLog's log collection technology is compatible with most SIEM and log analytics products, and can handle data sources. It provides Windows log collection capabilities, secured and reliable collection and transfer, remote deployment of configuration changes and monitoring agents, supports agent-less and agent-based log collection modes, support for a wide range of data formats and protocols, etc.



Step 3: Log Storage

After log collection and log transmission, the log data has to be stored in a particular place for analysis and auditing in the future. The log files collected from various devices, like antimalware tools, proxies, firewall, authentication servers, routers, and switches, should be stored in a central repository/database.

The following are a few points that should be kept in mind while deciding log storage.

- **Storage duration:** The duration of time for which the data is stored is known as storage duration. Storage duration for logs can be different. It depends on the type of log that is being stored. If it is a long-term log and does not need instant analysis, then it has to be stored in the cloud. However, if it is a short-term log, then it has to be stored in distributed storage. Thus, there are two different types of storage systems in log storage: cloud storage and distributed storage systems.
 - **Cloud storage:** This type of storage is used to archive and store long duration logs. This is flexible and relatively cheap for a large amount of data. It provides encrypted storage that keeps the logs secure during data transfer. In this storage, data is arranged in an indexed form.
 - **Distributed storage system:** This type of storage is used to archive and store short-duration logs that need to be analyzed frequently. The usage of distributed storage system requires physical equipment and occupies more amount of space, which is not cost-efficient.
- **Volume of the data to be stored:** The log data produced by each device may vary in memory as different devices produce different amounts of data. Therefore, different systems are used for applications that generate different amounts of logs. The generation

of log data depends upon the number of servers on which the application is running. A storage system that is used for storing logs should be highly scalable and must be able to function properly even with a large amount of data. It should be able to handle data growth over time.

- **Way of accessing the logs:** The storage system for storing the log data is to be selected based upon the frequency and ease of access. If the network defender wants to access the log data quickly, then distributed storage system or local storage is a better option. By default, the log viewer shows the log data, including syslog. The individual logs can be found by searching manually. If access to these files is difficult and does not fulfill monitoring purpose, then some storage systems cannot be used for real-time analysis.

Step 4: Log Normalization



1 Log normalization is the process of accepting logs from heterogeneous sources with different formats and converting them into a common format

2 During normalization, raw log data is collected from different sources and proper parsing expression is used to normalize the data. The logs are mapped with the standard scheme or framework to parse the data. Most of the log analysis systems use a regular expression to parse the data

3 Log messages are categorized into a more meaningful, predictable, and consistent piece of information after normalization

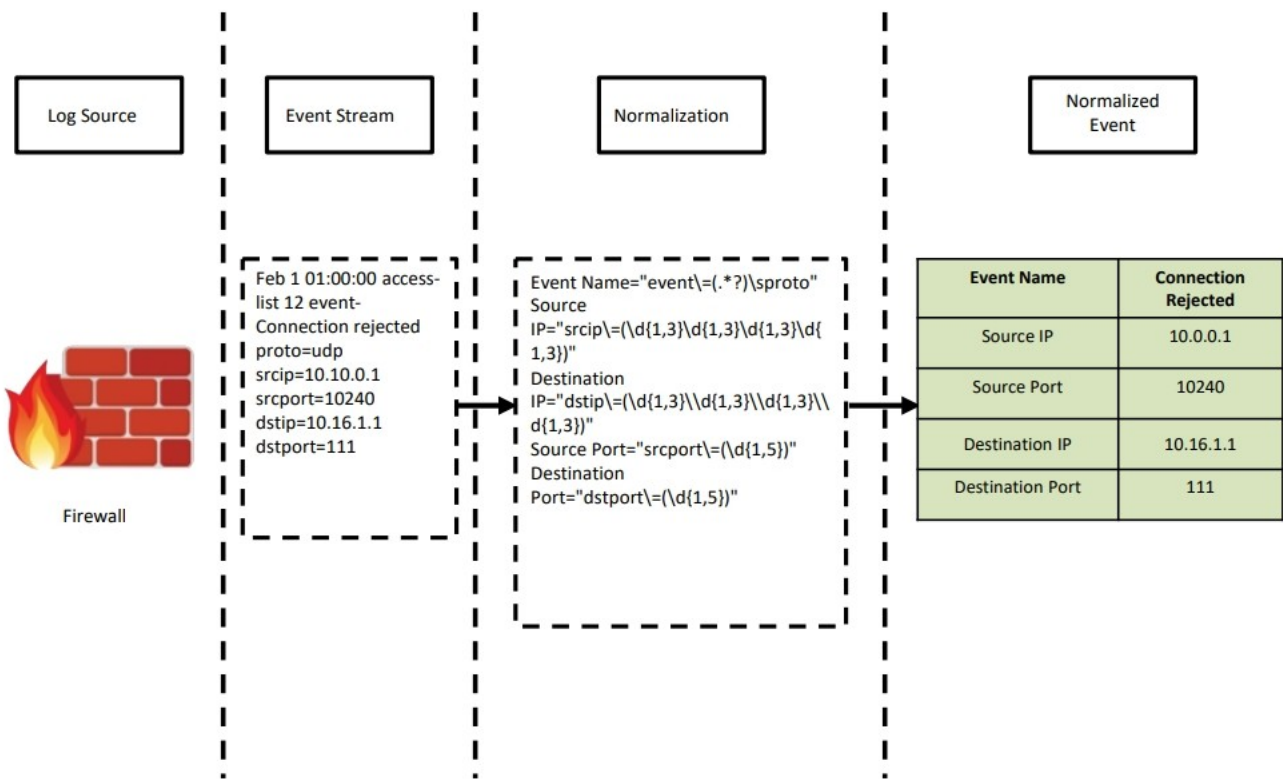
Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Step 4: Log Normalization (Cont'd)



Log Normalization Steps:

- The log collector collects logs from various sources
- The source type is identified based on the event
- The parser is loaded, and regex is set to identify the fields in the event
- The normalization is done, and the logs are categorized
- Aggregation and filtering are applied
- The same is repeated for each event



Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Step 4: Log Normalization

The various devices and applications produce log data in their own default format. For example, web proxy logs include source IP address URL, status code, browser name, and version, etc. Antispam logs include sender and destination email addresses, source IP address, source domain, spam score, etc. Similarly, firewall logs include source and destination IP addresses and ports, protocol, etc.

Collecting all this data in its different formats and then arranging and indexing it is a difficult task. Therefore, log normalization is needed to rectify this problem. Log normalization is the process of accepting logs from heterogeneous sources with different formats and converting them into a common format. It is performed regardless of the source and protocol used (i.e., syslog, SNMP, database, etc.). It forms an important step in log correlation.

During normalization, raw log data is collected from different sources and proper parsing expression is used to normalize the data. According to Common Event Expression (CEE), the logs are mapped with a standard scheme or framework to parse the data. Most of the log analysis systems use a regular expression to parse the data. Log messages are converted into a more meaningful, predictable, and consistent piece of information after normalization.

Steps Involved

The following steps describe the log normalization process:

- The log collector collects logs from various sources
- Source type is identified based on the event
- Parser is loaded, and regex is set to identify the fields in the event
- Normalization is performed, and the logs are categorized
- Aggregation and filtering are applied
- The above is repeated for each event

The below diagram represents the normalization process of a good event, where all normalized fields are highlighted in green:

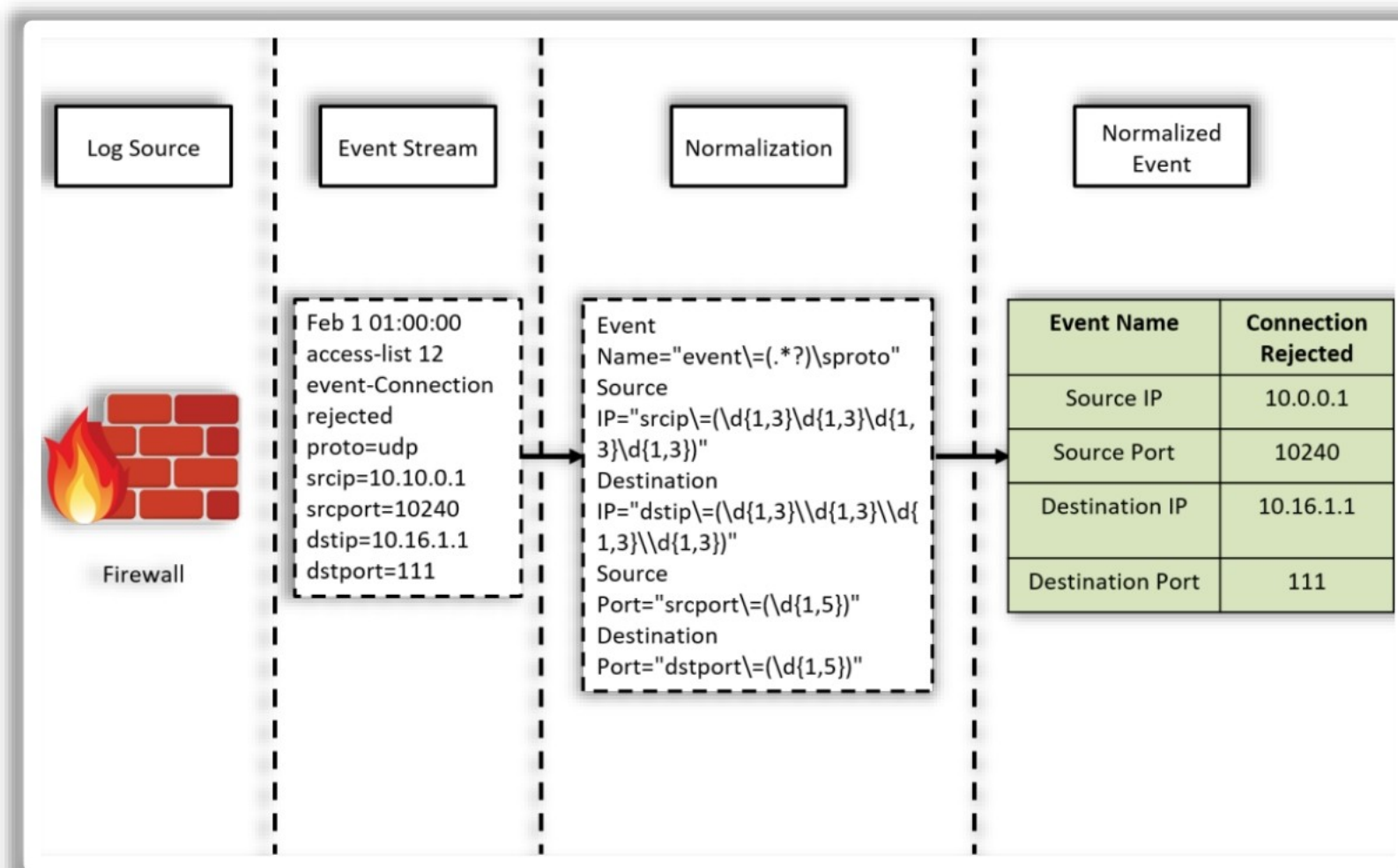


Figure 15.39: Normalization Process of a Good Event

The below diagram represents the normalization process of a bad event, where red highlighted fields could not be normalized because the parser missed some regex.

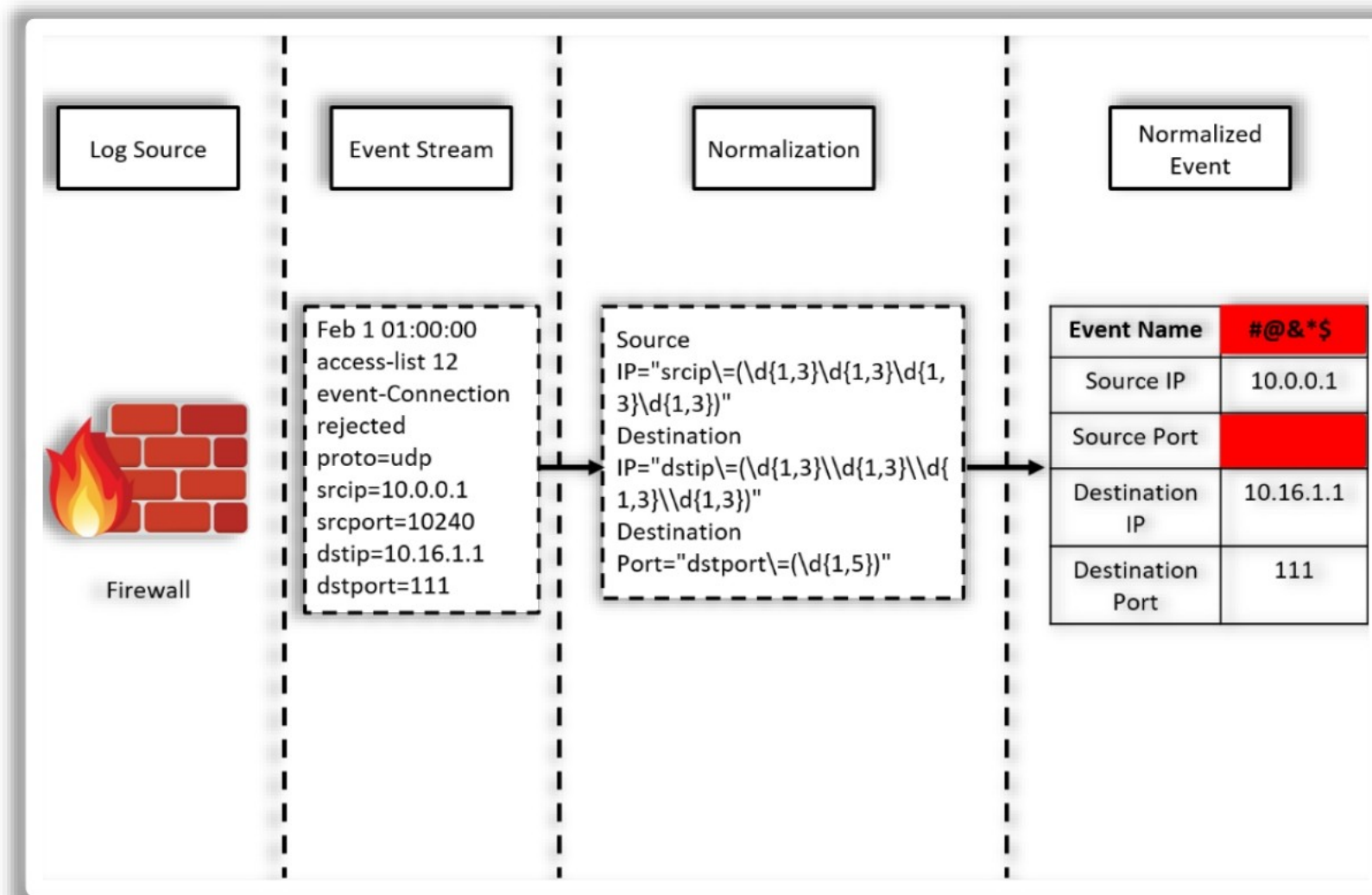



Figure 15.40: Normalization Process of a Bad Event

The following fields are common and should be considered for event normalization.

- **Source and destination IP addresses:** This field is used for log correlation.
- **Source and destination ports:** This field describes which services are accessed or going to be accessed.
- **Taxonomy:** It categorizes the meaning of log message.
- **Timestamps:** There are two types of timestamps: one describes the time when a specific log message was generated and the other describes the time when the log message reached the logging system.
- **User information:** This field describes username, command, directory location, etc.
- **Priority:** This field describes the priority of a specific message.

Step 5: Log Correlation



- Log correlation is the process of **matching a series of normalized log data** to determine a set of related events based on a certain set of rules
- It uses **rule-based correlation**, **statistical** or **algorithmic correlation**, and other methods to relate different events to each other
- When an incident occurs, these correlated logs are analyzed, and the cause for the incident is identified

Types of Correlation

Micro-level Correlation

- It correlates fields within a single event or set of events. It is also known as atomic correlation. It comprises field correlation and rule correlation

Macro-level Correlation

- Macro-level correlation gains information from rule correlation, vulnerability correlation, profile (fingerprint) correlation, anti-port correlation, watch list correlation, and geographic location correlation to validate and gain intelligence on the event stream

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Step 5: Log Correlation

Log correlation is the process of matching a series of normalized log data to determine a set of related events based on a certain set of rules. It uses rule-based correlation, statistical or algorithmic correlation, and other methods to relate different events to each other. When an incident occurs, these correlated logs are analyzed, and the cause for the incident is identified. However, the correlation of logs is very critical and complicated process due to the following reasons:

- Most logs are written in human-understandable, plain language, while others are scripted in cryptic languages with only esoteric system codes.
- Some systems have their siloed lenses. They look at their logs through an inefficient and incomplete filter. For example, a network IDS looks only for packets and streams; similarly, an application only looks for sessions, users, and requests.
- Logs are static; they will not contain all the context of ongoing events. Therefore, proper log analysis is required to understand the full context of an event.
- Comparison of the logs of an event from one system with the logs of the same event of another system of the same version may not give the same information.

There are two types of correlations: micro-level and macro-level.

- Micro-level correlation:** It correlates fields within a single event or set of events. It is also known as atomic correlation. Micro-level correlation is performed only when raw event data is normalized. It is further divided into field correlation and rule correlation.
 - Field correlation:** It is a mechanism through which tasks of interest can be found within the normalized event data. For example, to search for the events targeting the

Page 2380

Certified Network Defender Copyright © by **EC-Council**
All Rights Reserved. Reproduction is Strictly Prohibited.

- web server, focus on events that have port 80 or 443 as the destination port. Field correlation is also possible through event types.
- **Rule correlation:** In rule correlation, custom rules are used based on stateful behavior, counting, timeout, rule reuse, language, the priority of an activity, action to take on the event, etc.
 - **Macro-level correlation:** Macro-level correlation is also known as fusion correlation. It pulls in different sources of information such as rule correlation, vulnerability correlation, profile (fingerprint) correlation, anti-port correlation, watch list correlation, and geographic location correlation to validate and gain intelligence on the event stream.
 - The rule correlation in macro-correlation is similar to rule correlation in micro-correlation. The only difference is that rule correlation in micro-level correlation can be converted to a rule in macro-level correlation.
 - Vulnerability correlation is used to scan system vulnerabilities, which helps the management in increasing the security level of their systems.
 - Profile (fingerprint) correlation utilizes banner snatching, OS fingerprints, remote port scans, and vulnerability scans to collect information. It provides deep insights about the attackers and helps in remediation post an attack.
 - Anti-port correlation utilizes open port information to identify attacks in the slow or low category.

Step 6: Log Analysis



- Log analysis is a process of **identifying the patterns** and **anomalies** in the correlated log data that signifies the activity of any intrusion attempt or policy violation
- An **intelligent decision** is made based on patterns and anomalies found in log data to identify and confirm the incident
- Log analysis can facilitate system troubleshooting, forensics, security incident response, and effective management of applications and infrastructure

Log Analysis Can Facilitate:

- Checking whether internal policies, regulations, and audits are being followed or not
- Identifying and resolving security incidents occurred
- Troubleshooting systems, computers, or networks
- Identifying user behavior
- Performing security event forensics in incident investigation
- Identifying a change in pattern of logs that may indicate an incident
- Enhancing security awareness

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Step 6: Log Analysis

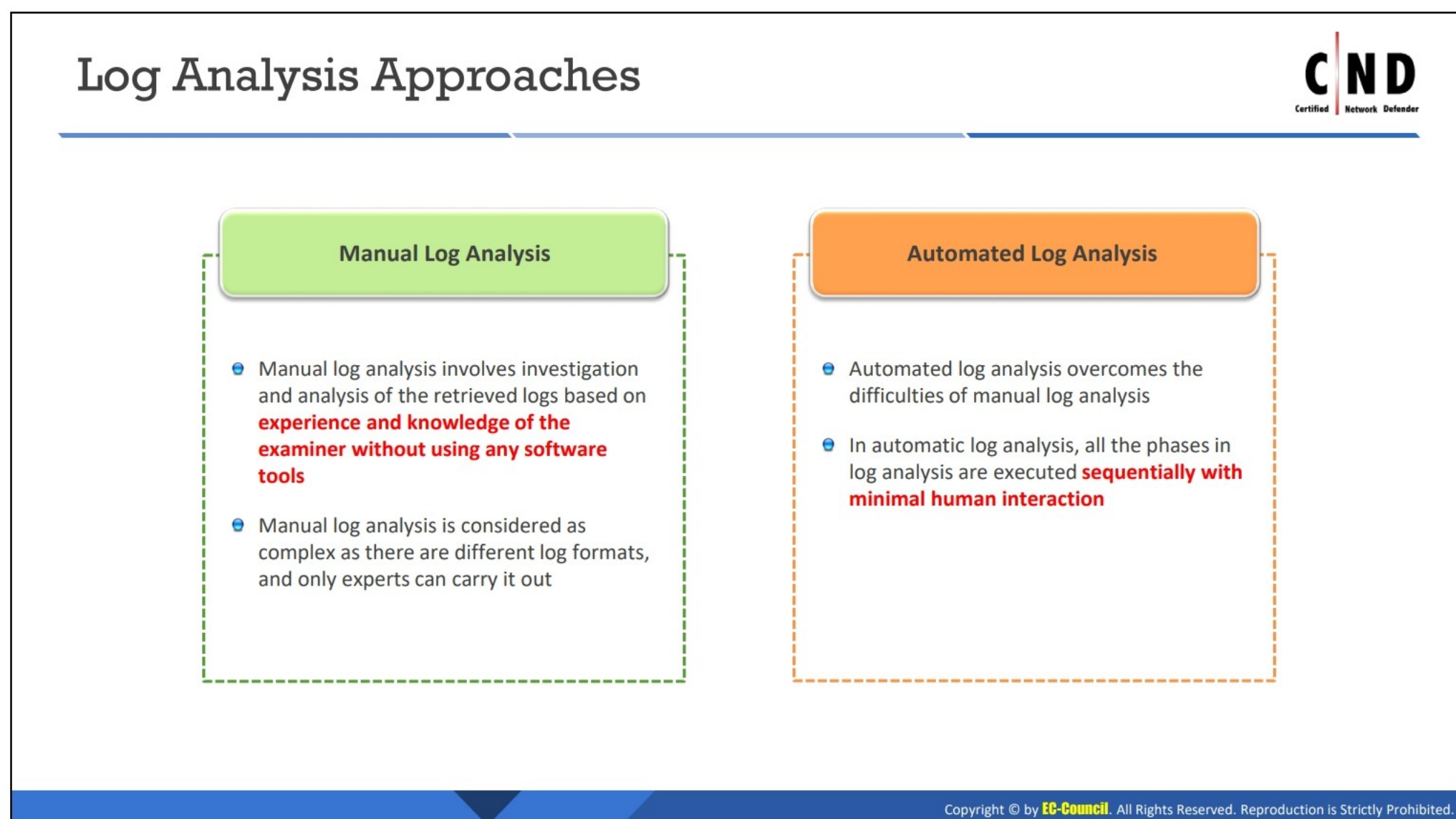
Logs are meant for analysis and analytics. Once the log data is stored, normalized, and correlated in a central location, it needs to be analyzed.

Log analysis is the process of analyzing correlated log messages to gain an in-depth understanding about them. It identifies the patterns and anomalies in the correlated log data that signify any intrusion attempt or policy violation activities. Intelligent decisions are made based on patterns and anomalies found in log data to identify and confirm the incident. It facilitates system troubleshooting, forensics, security incident response, and effective management of applications and infrastructure. It also identifies relevant events from the huge cluster of data and ignores the irrelevant ones. It enhances security awareness and supports in the identification of failed processes, protocol failures, or network outages. It also helps in identifying trends and upgrading search functionalities and performance.

During log analysis, a huge amount of log data is analyzed and labeling. Through labeling, it becomes easy for network defenders to monitor different event logs in a distributed and detailed manner in one place.

Log analysis can facilitate:

- Checking whether internal policies, regulations, and audits are being followed or not
- Identifying and resolving security incidents
- Troubleshooting systems, computers, or networks
- Identifying user behavior
- Performing security event forensics in incident investigation
- Identifying a change in the pattern of logs that may indicate an incident
- Enhancing security awareness



Log Analysis Approaches

The log analysis process is performed using two approaches. The first approach is a manual log analysis and the second one is automated log analysis.

Manual log analysis: In manual log analysis, a person manually goes through the log data that is collected, monitors it, and tries to identify any suspicious incidents happening in the network without using any software tools. It involves investigation and analysis of the retrieved logs based on the experience and knowledge of the network defender. Therefore, network defenders should have proper knowledge of the system; they should check whether every part of the system is operating normally or not; and they should understand what is going to be changed most recently.

Automated log analysis: Automated log analysis overcomes the difficulties faced in manual log analysis. In this approach, all the log analysis phases are executed sequentially with minimal human interaction. This identifies the patterns and anomalies in the correlated log data that signifies the activity of any intrusion attempt. This method of log analysis is highly preferred as it is time- and cost-efficient.

Log Analysis Best Practices



- Log analysis system should be synchronized with the NTP server to avoid timing differences between the systems
- Log analysis should always be considered as a proactive security initiative rather than a reactive one as it is often performed after an incident has occurred
- Automate the log analysis process as it takes less time and minimizes human interaction
- Review and analyze the logs at regular intervals
- Develop a baseline to detect unusual activities/events quickly when logs are registered

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Log Analysis Best Practices

Below are some of the best practices that should be used for log analysis:

- Log analysis system should be synchronized with the NTP server to avoid timing differences between the systems.
- Log analysis should always be considered as a proactive security initiative rather than a reactive one as it is often performed after an incident has occurred.
- Automate the log analysis process as it takes less time and minimizes human interaction.
- Review and analyze the logs at regular intervals.
- Develop a baseline to detect unusual activities/events quickly when logs are registered.
- Set a strategy to log data.
- Set an effective logging format to detect and gain an understanding of the network from the logs.
- Collect the logs at a centralized location, separated from the production environment.
- Implement end-to-end logging to gain a more holistic view.
- Correlate data sources to detect events that cause system malfunctions.
- Use unique identifiers for debugging, support, and analytics.
- Always perform real-time monitoring.

Step 7: Alerting and Reporting



An alerting system in a centralized logging application alerts the user if any suspicious event is observed in the logs or calculated matrices

Purpose of the Alerting System:


- Error reporting
- Monitoring

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Step 7: Alerting and Reporting

An alerting system will alert the user if any suspicious event is observed in the logs or calculated matrices. It is necessary that a centralized logging system should have an alerting system that monitors logs for any changes and send a notification if any abnormalities detected. The notification may take place in many ways (email, desk tickets, etc.); however, the respective staff should be informed in due time, so that they can take precautionary measures to prevent the breach. This alerting mechanism helps the organization not only improve its security but it also gives a 360-degree view of the activities that are going on in the network.

Centralized Logging Best Practices



- 1 Ensure **logging feature is enabled** on the network devices
- 2 Consult the **legal department** when developing policies regarding storage, retrieval, analysis, etc. of the log files
- 3 Ensure safe **transmission** and **storage** of the logs in the network
- 4 Consider **all sources of logs** in the network and collect appropriate logs


Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.













Centralized Logging Best Practices

Below are some of the best practices for centralized logging:

- Ensure that the logging feature is enabled on the devices that are connected to the network.
- Administrator should be able to quickly handover the authority to security professionals to give access to them at the time of an emergency.
- Consult the legal department when developing policies regarding storage, retrieval, analysis, etc. of the log files.
- Ensure safe transmission and storage of the logs in the network.
- Consider all sources of logs in the network and collect appropriate logs.
- The data that is stored should be accessible when investigating an incident.
- Authentication and security should not be compromised in the process of making the data available.
- Maintain a consistent structure of the logs that are being stored.
- Set the severity levels for the alerts.
- Indexing and storing incident logs is a must for future reference and performing correlation using that data.

Centralized Logging/Log Management Tools



 Splunk https://www.splunk.com	 Retrace https://stackify.com
 Logmatic https://logmatic.io	 Logentries https://logentries.com
 Logstash https://www.elastic.co	 Graylog https://www.graylog.org
 Sumo Logic https://www.sumologic.com	 Xpolog http://www.xpolog.com
 Papertrail https://papertrailapp.com	 LOGalyze http://www.logalyze.com
 LogRhythm http://logrhythm.com	 Loggly https://www.loggly.com

Copyright © by **EC-Council**. All Rights Reserved. Reproduction is Strictly Prohibited.

Centralized Logging/Log Management Tools

Below are some CLM tools that can be used for centralized logging:

- **Splunk**

Source: <https://www.splunk.com>

Splunk aggregates and analyzes log data. It provides insights to quickly detect and respond to internal and external attacks and simplify threat management. It helps teams gain organization-wide visibility and security intelligence for continuous monitoring, incident response, security operations, and provides executives a window into business risk.

- **Logmatic**

Source: <https://www.datadoghq.com>

Logmatic is a log analyzer tool that automatically detects unexpected behavior that would have previously taken days to find using traditional log-processing tools. It provides very granular information that reduces the bug fix time.

- **Logstash**

Source: <https://www.elastic.co>

Logstash is an open-source, server-side data processing pipeline that ingests data from a multitude of sources, simultaneously transforms it, and then sends it to the preferred "stash." It easily ingests from the logs, metrics, web applications, data stores, and various AWS services, all in continuous, streaming fashion.

- **Sumo Logic**

Source: <https://www.sumologic.com>

Sumo Logic is a suite of applications and integrations. It is used to tackle common cloud infrastructure challenges such as log management, real-time monitoring, resolving user experience issues, and performance issues for major cloud platforms. Sumo Logic can help analyze and correlate AWS CloudFront data with the origin data/other data sets and improve availability and end-user experience while enforcing rigorous security controls.

- **Papertrail**

Source: <https://papertrailapp.com>

Papertrail provides log management tools for search, live tail, flexible system groups, team-wide access, and integration with popular communication platforms such as PagerDuty and Slack to track down customer problems quickly, or troubleshoot slow database queries. It can be used with a wide variety of log types, including syslog, text log files, Apache, MySQL, Ruby on Rails, Windows events, Tomcat, routers, firewalls, etc.

- **LogRhythm**

Source: <http://logrhythm.com>

LogRhythm is an end-to-end platform that is designed by security experts. LogRhythm delivers patented, high-performance, distributed, and highly available processing of machine and forensic data received from data collectors, system monitors, and network monitors and then transforms it into a contextualized form.

- **Retrace**

Source: <https://stackify.com>

Retrace is an application performance monitoring solution designed for developers to improve code, which in turn improves performance and fixes hidden exceptions. It gives developers all the application insights they need in one place.

- **Graylog**

Source: <https://www.graylog.org>

Graylog is designed for log collection, storage, enrichment, and analysis. It offers simplicity in searching, exploring, and visualizing data, which means no expensive training or tool experts are required. Graylog performs speed analysis, provides a more robust and easier-to-use analysis platform, and offers simpler administration and infrastructure management.

- **XpoLog**

Source: <http://www.xplg.com>

XpoLog is used to discover errors and problems in log data. XpoLog filters the search results and uses a complex search syntax, which results in a summary table of events and transactions, including insights for further investigation. It allows creation of custom rules


in a predefined layer and it automatically layers the auto-detected layer. This way, all rules are covered in the search.

- **Loggly**

Source: <https://www.loggly.com>

Loggly 3.0 charts provides a variety of ways to quickly visualize data, and its dashboards helps organize this data in the most useful ways for detecting and understanding the problems that arise in software and infrastructure.

Centralized Logging Challenges



1

Existence of **multiple log sources** due to many hosts throughout the organization

2

Different log sources **generate logs of different log format**, which makes difficult to review

3

Managing the available resources with continuously **increasing log data**

4

With the changing threat landscape, it is **difficult to** monitor using existing capabilities

5

Difficulty in **determining** the purpose and importance of data sources

6

The **timestamp** of every log is set using its internal clock. (If the host's clock is incorrect, then it makes it difficult to analyze the logs and even more complicated when the logs are collected from multiple hosts)

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Centralized Logging Challenges

Organizations face various log management-related challenges that are categorized into three parts: first, challenges in generation and storage of logs due to their variety and prevalence; second, challenges in maintaining confidentiality, integrity, and availability of logs; and finally, challenges in identifying skilled people for performing log analysis.

- **Challenges in log generation and storage:** Various hosts, OSes, security system, and applications generate and store a variety of log files, which makes log management a complex process.
 - **Many log sources:** Multiple log sources are present on many hosts throughout the organization, and a single log source may produce several logs. For example, a web application may store network-related activities in one log and authentication-related activities in another log.
 - **Inconsistent log content:** A single log source records a specific portion of information in its log entries such as host IP addresses, usernames, etc. Further, for maintaining efficiency, the various log sources store only that portion of the information that is most important to them. Due to this, it becomes difficult to connect events generated by different log sources as log entries recorded by one log source may be different from the log entries recorded by another log source, and they would hardly have any common values. For example, log source A may record username but not the host IP address whereas log source B may record host IP address but not the username.
 - **Inconsistent timestamps:** The timestamp of each and every log is set using its internal clock. It becomes difficult to analyze the logs and even more complicated when the logs are collected from multiple hosts if the host's clock is set incorrectly. For example,

an incorrect timestamps could display that event M occurred 30 s before event N. However, in reality, event M may have occurred 1 min after event N.

- **Inconsistent log formats:** Different log sources generate logs in different log formats such as databases, tab-separated or comma-separated text files, XML files, and binary files. Some logs use standard formats, and others use proprietary formats. Some are developed to store locally, while others are developed for transmission to another system. This complicates the process of log review. Therefore, to avoid such challenges, organizations need to adopt automated techniques for converting logs with different formats into a standard format.
- **Challenges in protecting log:** Log entries should be secured against any breach of confidentiality or integrity. Logs collect critical and sensitive information such as login credentials, emails, etc. intentionally and unintentionally. This increases the security and privacy concerns for both, the persons that are going to perform the log review and to others that may access logs by authorized or unauthorized methods. If the logs are not secured properly, they are susceptible to alteration and destruction. This leads to a variety of security issues such as permitting malicious activities without any warnings and alerts, exploiting evidence to hide the identity of a suspicious person, etc. Further, the availability of logs also needs to be protected. Logs have a fixed size to store log data. If that size reaches its maximum limit, then the log will overwrite the old data with the new data, making old log data unavailable. To overcome this problem, save copies of log files for a longer period of time than the original log sources can support.
- **Challenges in log analysis:** Generally, system administrators are responsible for performing log analysis to determine events of interest. However, this task is considered as a low-priority one by most system administrators as they are responsible for managing operational issues as well as providing solutions to security risks and vulnerabilities. Moreover, they may not have received any training to perform log analysis efficiently or may not have any tools to automate the process of log analysis. Some system administrators may not find the process of log analysis interesting or a productive use of their time. As mentioned earlier, log analysis needs to be considered as a proactive process instead of a reactive one. Suspicious activities or indicators of compromise should be detected before any critical problems arise.

Module Summary



- Logs play a pivotal role in incident detection
- Almost every device on the network has the capability to produce logs
- The log file contains various types of information that help provide valuable and actionable information
- Monitoring and analyzing log files of different devices locally can be a difficult task. Centralized logging helps you to simplify the process
- In centralized logging, logs from different devices and applications on the network are collected to one central location
- Centralized logging, monitoring, and analysis are done through a series of steps, which includes log collection, log transmission, log storage, log normalization, log correlation, log analysis, alerting, and reporting

Copyright © by EC-Council. All Rights Reserved. Reproduction is Strictly Prohibited.

Module Summary

In this module, we discussed incidents, events, and logs; log sources that generate logs; the role of logging in detecting security threats; and local logging and CLM concepts. In local logging, we discussed monitoring and analysis of different logs of various devices such as Windows logs, Linux logs, Mac logs, firewall logs, Windows Defender Firewall logs, Mac OS X firewall logs, Linux firewall logs, Cisco ASA firewall, Check Point firewall, router logs, web server's logs, etc. This module also discussed monitoring and analysis of logs through centralized logging.