Chapter 14
Log Analysis

Learning Objectives
The objectives of this chapter are to:

- Know two popular logging mechanisms, Syslog and Windows Event Log, and understand how they work
- Know how to configure syslog
- Be able to collect, parse and analyze logs
- Understand SIEM works

The preceding chapters have mainly focused on the most common source of digital evidence, computer storage devices. Another important source of digital evidence is log files. One of the keys to success in conducting an effective digital investigation on a computer system is to know what is happening on the system. Computer systems and applications generate logs when something happens or needing for attention (e.g., a computer system being configured to record user login attempts.). These logs can provide solid forensic evidence to reveal a user's misbehaving activities and discover how, when and where of an incident and help identify cybercriminals. In this chapter, you'll learn two major logging mechanisms, Syslog and Windows Event Log, and how they work. Also, you'll learn Security Information and Event Management System (SIEM). Finally, you'll know how to collect, parse, and analyze logs.

© Springer Nature Switzerland AG 2018

X. Lin, *Introductory Computer Forensics*,
https://doi.org/10.1007/978-3-030-00581-8_14

14.1 System Log Analysis

Computer systems and applications generate large amount of logs to measure and record information for the duration of the monitoring period. System log data is one of the most valuable, containing a categorical record of user transactions, customer activity, sensor readings, machine behavior, security threats, fraudulent activity and more. When security breaches happen, logs may be the best line of defense [1]. System logs are also one of the fastest growing, most complex areas of big data, especially with the rapid development of distributed computing. The large amount of logs, which is generated by various systems, are a very computationally intensive task for mining by analyzing them. Precisely mining and analyzing logs data will efficiently improve system security, strengthen system defense capability, and attacks forensics.

Windows Event Log and Linux/Unix syslog are the two major logging mechanisms. Usually, they are deployed simultaneously in the complex network physical environment for log management, and both of them can be custom configured by the end users. There are tools available to integrate Windows Event Log and Linux/Unix syslog to make log management more efficient in today's enterprise environment. For example, SolarWinds Event Log Forwarder for Windows is able to automatically forward Windows event logs as syslog messages to a syslog collector [2]. However, some common analytics challenges are caused by traditional log data management and limited by current technologies. Along with some of these challenges, which includes log generation, collection, transport, storage, analyzing and forecasting in the perspective of security will be presented in this section.

14.1.1 Syslog

Syslog (System Logging) is a standard for message logging, and is widely used on Unix and Linux as well as many security products, such as Firewall, Intrusion Detection System (IDS). The syslog log messages are classified by two categories: Facility and severity [5]. The "severity" is used to indicate the priority, which is shown in the following table (Table 14.1).

Table 14.1 Syslog message severities

Numerical code	Severity
0	Emergency: system is unusable
1	Alert: action must be taken immediately
2	Critical: critical conditions
3	Error: error conditions
4	Warning: warning conditions
5	Notice: normal but significant condition
6	Informational: informational messages
7	Debug: debug-level messages

Table 14.2 Syslog message facilities

Numerical code	Facility	Description
0	kern	the kernel
1	user	"user" processes (no specific)
2	mail	sendmail
3	daemon	"system" daemons, such as 'routed'
4	Auth	security and authorization-related
5	syslog	Syslog internal messages
6	LPR	BSD line-printer daemon
7	news	usenet news system
8	UUCP	for the (ancient) UUCP (unix-to-unix copy) service
9	Cron	the cron daemon
10	authpriv	similar to 'auth', logged to secure file
11	ftp	FTP daemon
16–23	local0–local7	used for local/other daemons

The "facility" describes the part of the system or application which generated the log message, which is shown in the following table (Table 14.2).

When a user tries to log into a Linux machine, for example, by using ssh (secure shell), the user is authenticated by entering his/her username/password. Then, an authentication event will be logged no matter whether it is a successful or failed login attempt. The followings are some examples of syslog messages of login successes/failures:

Oct 16 17:10:30 localhost sshd[5124]: Accepted password for root from 192.168.220.1 port 1643 ssh2

Oct 16 17:10:31 localhost sshd[5127]: pam_unix(sshd:session): session opened for user root by root(uid=0)

Oct 16 17:10:51 localhost sshd[5154]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.220.1user=root

Oct 16 17:10:53 localhost sshd[5154]: Failed password for root from 192.168.220.1 port 1645 ssh2

Oct 16 17:10:59 localhost sshd[5154]: Failed password for root from 192.168.220.1 port 1645 ssh2

Obviously, if we can keep monitoring those log messages, it is likely that we can detect many attacks/abuses. For example, if we find many failed login attempts against a user account in a short period, it is highly like that the system is under password guess attack.

Although the syntax and semantics of data in log messages are usually vendor-specific, all of them should follow the syslog protocol (RFC 5424). The protocol utilizes a layered architecture, which allows the use of any number of transport protocols for transmission of syslog messages. Without this protocol, each other standard needs to define its own syslog packets format and transport mechanism, which may cause subtle compatibility issues. Syslog protocol defines three layers,

Fig. 14.1 Syslog mechanism processing flow

- syslog content—the management information contained in a syslog message
- syslog application—handles generation, interpretation, routing, and storage of syslog messages
- syslog transport—puts messages on the wire and takes them off the wire

Syslog processing flow can be generally revealed by several stages. Firstly, source objects information must be collected. After raw data being filtered by the vender-specific mechanism that usually is regular expression, the events information should be written into log text massage. Meanwhile, if log files will be archived or transferred, the further activities should be logged with destination information as well (Fig. 14.1).

When syslog was first introduced, it only supported UDP for log message delivery. It means that there is no guarantee that the log messages will be successfully delivered to its predefined destination(s). Later, enhanced versions of the syslog protocol have emerged as promising logging mechanisms for a wide range of computing devices today, having more functionality than their ancestor. For example, Syslog-ng (Syslog Next Generation) extends basic syslog protocol with new features including content-based filtering, logging directly into a database, reliable transport using TCP and secure transmission using TLS (Transport Layer Security) [6]. Another enhanced version worth mentioning is rsyslog, and the most notable enhancement by rsyslog is its high-performance and great security features [7]. Nowadays, many Linux distributions have pre-built package of either Syslog-ng or rsyslog available. For example, Kali Linux used in our book has rsyslog package installed. Hereafter we will use rsyslog for log analysis.

Sample log collection deployment scenarios using syslog work like the following: Log messages are generated by an 'originator' and forwarded on to a 'collector'. The syslog collector is usually a centralized logging server or service for centralized logging and event management.

Centralized Logging has many advantages. First, it allows logs from different systems to be checked on a single system, and as a result it might become easier to find out the root cause of incidents. Most importantly, it still provides trail when the originator is compromised. This is because that it is very common that a hacker always clears log files after having done something on the compromised computer system.

14.1.1.1 Configuring and Collecting Syslog

On UNIX and Linux, syslog includes a configuration file [8]. The default configuration file for syslog, rsyslog and syslog-ng are /etc/syslog.conf, /etc/rsyslog.conf

and /etc/syslog-ng/syslog-ng.conf, respectively. Note that only administrators with root permission can modify the configuration file.

While rsyslog is an "advanced" version of syslog, its config file, "rsyslog.conf", remains the same as the one used by syslog. In other words, if you copy a "syslog.conf" file directly into "rsyslog.conf", it still works. However, /etc/syslog-ng/syslog-ng.conf has a totally different structure compared to the other two.

The configuration file indicates what logs and where to save. It is a text file, and every line in this file is called a rule. We take /etc/rsyslog.conf as an example, and each line or rule has the following format

selector <Tab> action

Specifically, the selector selects what logs will be recorded and saved, whereas the action describes how logs will be saved. It means rules map selectors to actions, which allows the Linux system logging facility (here rsyslog daemon) to send messages of certain types to different locations. Note that lines that start with "#" are comments and blank lines are ignored. Multiple selectors with the same action can be combined with a semicolon. Also, it is worth mentioning here that selector and action are separated by a TAB character, not a whitespace character.

The "selector" has the following format

facility.priority

where facility indicates the program sending the message or whose log and priority indicates the severity level of the message or what log. Special values can be used in a selector. For example, * stands for all possible values, and none stands for no priority of the given facility. Note that message is logged if its priority is at least as severe as the priority specified. Also, multiple facilities with the same priority can be separated by commas. For example, you might want to log anything (mail and authpriv) of level info or higher, using the selector "mail,authpriv.info". Common facilities include user, kern, mail, daemon, auth, lpr, news, uucp, and cron. The severity levels listed from most importance to least important are: emerg, alert, crit, err, warning, notice, info, debug, and none.

The action describes how messages will be logged, including log files, console, and remote hosts.

An example of the format would be:

auth,authpriv.*<Tab>/var/log/auth.log

It means all the user authentication messages including login logs are written to a file named auth.log in the folder of /var/log.

Next, we take a look at how to put user authentication messages into /var/log/forensics.log, by doing the followings:

(a) Log into Forensics Workstation as root
(b) Change into /etc
(c) Edit "rsyslog.conf" by using an editor, for example, vi or emacs, and add the following line in the "rsyslog.conf" file

    auth,authpriv.*<Tab>/var/log/forensics.log

Note that the separator used in the above line is TAB.

(d) Restart rsyslog service
/etc/init.d/rsyslog restart

Note that restart of the rsyslog daemon is required to have the just added configuration active.

14.1.1.2 Viewing the Log Files

(a) Issue the following command
tail -f /var/log/forensics.log
(b) Generate some logs by logging into Forensics Workstation with both correct and wrong passwords. You should see output that show your login activities.

This is an example for audit event, which records a failed attempt as well as successful login/logoff of user with UID of 0 to log in as the root user (Fig. 14.2).

In this example, we can clearly see that a user log-in or log-off event will generate many messages. Unfortunately, it becomes a challenge to us when we analyze system logs.

14.1.2 Windows Event Log

Unlike UNIX syslog Windows system logs are structured. Logging on Windows system is viewable through the 'Event Viewer'. Event logs differ for different variants, but windows 10 made important improvements on security. In the console tree, open Windows Logs, and then click Security. The results pane lists individual security events, which shows as Fig. 14.3. Aside from basic logging on system

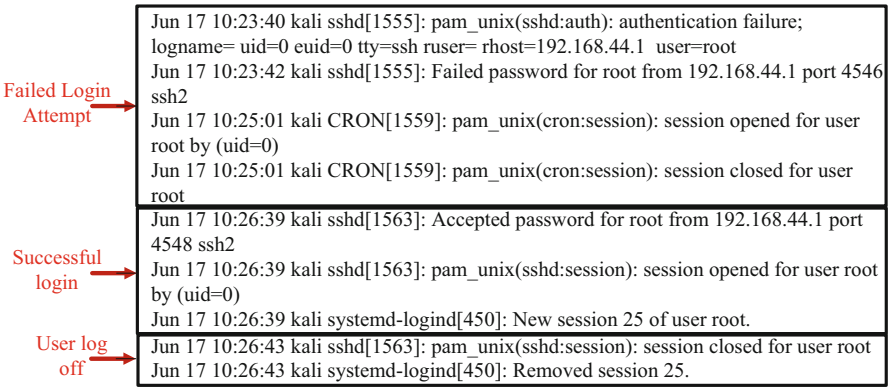


Fig. 14.2 Audit event logs of a failed attempt login and a successful login/logoff

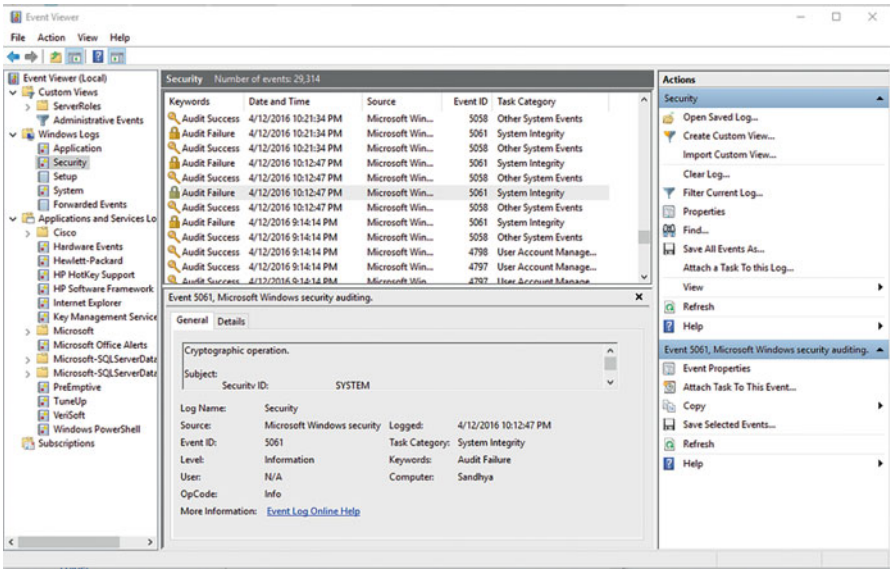


Fig. 14.3 Security logs on Windows 10 system

crashes, component failures, logging in and out, system file accesses, security logging also can covers application and per-application logs that are requested by applications.

The current Windows logging infrastructure of desktop and server operating system is composed of TraceLogging and Event Logging framework [4]. The new released framework of TraceLogging builds on Event Tracing for Windows (ETW) and provides a simplified way to instrument code. However, the security log, which is designed for use by the system, generates in Event Logging framework.

The system grants access based on the access rights granted to the account under which the thread is running. Users can read and clear the Security log if they have been granted the SE_SECURITY_NAME privilege, and only the Local Security Authority (Lsass.exe) has write permission for the Security log, and no other accounts can request this privilege. Logging for the authentication events category must enable the AuthzReportSecurityEvent function to generate logs. And audit parameters can be custom defined in AUDIT_PARAM_TYPE function as well (Fig. 14.4).

However, Windows cannot limit the privilege of administrator as SELinux [12, 13], the only way to prevent a Windows system from overwriting its log files is to configure it to stop logging when it runs out of disk space, or to shut the machine down entirely. Likewise, readable weakness of windows application log is apparent as well. In Windows event log configuration, programmers write a 'Message Dictionary' that maps coded error numbers that released by Microsoft. In such a case, code '80010105' from the log file into matching messages in a DLL, user may need to a helpdesk to know what an '80010105' is.

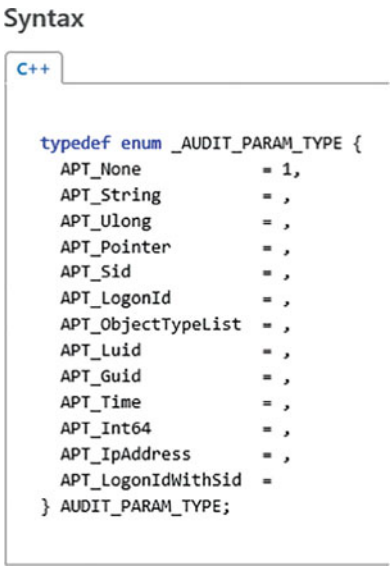


Fig. 14.4 Windows audit parameters type for authentication events

14.1.3 Log Analytics Challenges

Today, enterprise environments have become more complex with many computer systems and networks of different types. Also, they secure themselves with a variety of third party security technologies from a number of vendors distributed located in the different locations of the enterprise network. The complex environments pose challenges to log analysis.

First, computer devices in an enterprise environments provide an overwhelming amount of information. It becomes infeasible for human review. Further, log messages comes in different formats, and the format of logs from some devices can be also custom defined. And to make matters worse, some devices use specialized codes or signatures to identify what "type" of alert or error is being generated. For instance, in Windows NT, the "Event Viewer" has codes like 529, which represents login failure because of unknown user name or incorrect password.

Second, logs vary greatly from system to system, and even from version to version for the same system. Crossing data sources correlation to investigate into the root cause, discover behavioral connections, and exclude duplicate information among different events, is one of the tough bottlenecks of log data analysis. As a matter of fact, mining evidence for security forensics may be recorded in multiple log files, and even cross multiple devices. While discovering relations between multiple sources can increase in complexity when dealing with more than two sources. Most of current technologies simply send all of log data into one centralized location and through complex search query language to achieve preliminary correlation, which leads to high cost but with only little success.

Third, today's enterprises have adopted a variety of security products. Different products focus on providing point-solution technologies for known security problems. For example, in controlling access to internal company networks, there are Firewalls and VPNs, such as Checkpoint's FireWall-1; to keep computer resources and/or networks physically inaccessible to unauthorized people, there is physical security or biometric security, such as fingerprint scanners and fingerprint recognition; to detect possible intrusion, there are intrusion detection systems (IDS), such as Cisco Firepower NGIPS; to analyze data packets transmitted into and out of networks, there are sniffers, such as Sniffer Technologies' Sniffer; to mitigate the risks of virus attacks, there is anti-virus software; to ensure the integrity of system files and data across your network and receive timely detection and notification of changes to the system, there is Tripwire; to protect your confidential and sensitive data sent over public network, for example, Internet, there are cryptographic technology and security protocols, such as IPSec, SSL/TLS, etc.; while there are security scanners that enable the scanning and mapping of networks and associated vulnerabilities, and the analysis of security levels, which potential weak spots are checked, identified and analyzed. As well, operating systems (OS) can be configured to log and/or transmit security-sensitive events. Each type of security products protects against a class of risk or more. However, not all of security incidents can be detected correctly and effectively by a single security product without the help of the information from others. For example, an employee launches a DoS attack against one of his corporation's server inside the organization, we want to immediately know who attacks the server and whether it is a successful attack. This is impossible for us to know if we only depend on IDS system.

Finally but not the last point. Except various log formats, analyzing performance is another prime limitations for log data analysis. In the traditional logging solutions, index can be used to accelerate the logs search, while it cannot effectively ensure the query in a large scale log data, especially for real-time troubleshooting and forensics.

In terms of above statements, we can clearly see that log analysis system needs correlate the information from diverse sources in a manner that identifies the root cause of an incident and explain digital artifact. This is why Security Information and Event Management System (SIEM) comes into play.

14.2 Security Information and Event Management System (SIEM)

Security Information and Event Management System (SIEM) is also known as Enterprise Security Management (ESM) or Security Event Management (SEM), and Security Information Management (SIM). SIEM manages information collected from computers, network devices and security products in an enterprise, and these information is further correlated to find the connection between them and identify the root cause of an incident. Also, it allows for the prioritization of incidents and

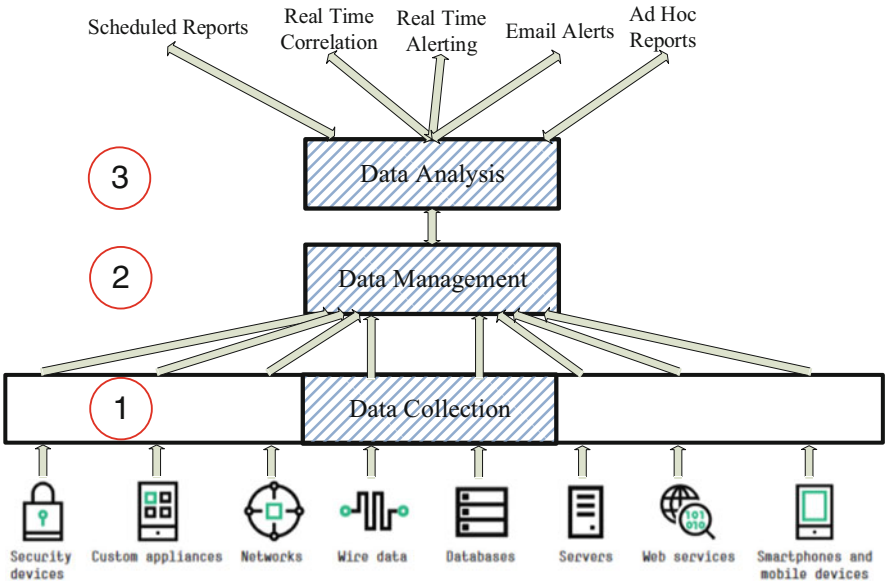


Fig. 14.5 SIEM architecture [14]

identify these which are critical to your business processes as well as your organization's security posture. In summary, SIEM manages information from computer devices in an enterprise in an efficient manner that enhances enterprise security.

The Security Information and Event Management System (SIEM) implies the ability to collect, process and correlate log messages from all monitored computer devices [11]. Variety of different logs formats and message types oblige the SIEM to correlate all actives and events in systems and even in networks. Generally, a typical SIEM Architecture can be divided into at least three procedures. As it's showed in Fig. 14.5, from log data gathering and standardizing, to log data management and analysis.

There are at least eight following features which an SIEM must have:

1. Collection: Centralized data collection and reporting for computer systems and network security products throughout the enterprise, ensuring that systems are in compliance with security policy and monitoring events from Firewall, IDS, web servers, Windows and Linux servers.
2. Classify: One of the problems with event logs and alerts generated by computer devices is that they use specialized codes to identify what "type" of error or alert is being generated. For, instance, in Windows NT, the "Event Viewer" has codes like 6005, 6009, 20, etc. But, it doesn't tell you that 6005 means "The Event Service was Started", or that 20 means "A file was printed". The products themselves do a bad job at presenting what is going on. Since no one can be an expert in everything, it makes it very hard for system administrators to understand what is reported by the great many devices in an enterprise. SIEM should provide

a new type of classification scheme, which is not based on a numeric code. It must be based on simple, human readable and easily understandable "types" such as, "ids.detect.dos", "auth.login.success" or "auth.login.failed", "auth.logoff".

3. Normalize: It translates various formats of raw log data into a standardized one. In doing so, log parser programs are included in the SIEM itself, and each of them is responsible for parsing raw log data from one specific computer device into the standardized format. Log Parser output files can be many types of formats. Currently, XML, SQL, and JSON are the most popularly used formats for log data analysis. Choosing the format that is fit for the organization is dependent on needs. XML is a very powerful format that allows a huge amount of flexibility in both the output of specific data and format. Log Parser can define use multiple XML structures and XML scheme. The SQL format allows users to convert log files data into a SQL table and store it in a relational database. JSON is derives from JavaScript, and it has been regarded as the best application format in recent years. JSON makes it possible to analyze logs like big data. It's not just readable text, but a database that can be queried.
4. React: Most monitoring products can "trigger" on a set of parameters or thresholds to alert system managers to pay attention. Typical notifications/reactions common to software include emailing, paging, running commands, or generating "troubleshooting tickets" in helping desk software like Remedy.
5. Event correlation: Event correlation can help us reduce alerts and identify points of security vulnerability across networks, systems and (more rarely) applications. Event correlation can help us detect pattern in a low-level stream, such as Syslog, SMTP, SNMP, etc., and generate "derived" higher-level event in order to reduce volume of traffic to SIEM systems [3]. Study shows that between 60% and 90% of the time IT managers spend resolving problems is lost to diagnostics. Event correlation promises to significantly reduce that percentage—bringing down operational costs for IT, and reducing revenue lost to downtime by many millions of dollars for large businesses. Also, a correlated knowledge base can improve accuracy and efficiency for other applications, such as those targeted at trending, performance and service-level management. In the most intricate of examples, information from a correlated, self-adaptive knowledge base can inform such actions as software distribution, configuration and change management—pushing towards pretty much the whole range of management disciplines.
6. Analysis: Frequent analysis of security relevant symptoms minimizes the risk of performance losses and security breaches.
7. Reporting: Flexible reporting provides decision support for different groups of people in an enterprise, including management and technical staff.
8. Security policy establishment: A well-conceived security policy is the foundation for true information security in any corporate computing environment. This policy, based on balancing risk versus cost, should concentrate on delivering integrity, availability and confidentiality. Implementing and measuring this policy in large, enterprise wide, multi-platform environments is an overwhelming task. SIEM gives you the ability to automate the planning, management and control of your security policy from a single location, thereby saving your time

and money. SIEM does this by giving you the ability to off load these repetitive and redundant tasks associated with managing such a policy to computers rather than relying on human staff members. Also, modern enterprises face many security management challenges. For example, new end users must have quick, easy access to a variety of distributed platforms and applications in order to be productive. User-access rights must also be instantly revocable to prevent unauthorized access and protect enterprise-wide security. Security procedures are designed to halt breaches; however, without timely centralized management, security procedures cannot be enforced. In addition, enterprises must keep pace with the proliferation of new IT resources, increasing numbers of users and the incorporation of new business channels such as e-business. Coordination, implementation and tracking of these procedures becomes increasingly complex in large enterprises with diverse, distributed IT infrastructure. SIEM approach combines a proven methodology and complete end-to-end services with the highly scalable technology. Its solution not only raises overall security levels, but also reduces day-to-day management tasks, offering enterprises the flexibility they require to compete in today's new economy.

Among the above features, log normalization, correlation, and analysis are some functions that determine how effectively a SIEM product conforms to the security needs.

14.2.1 Log Normalization and Correlation

Log correlation is trying to pull all information together, which will definitely be beneficial in getting clues of malicious incidents across multiple data source. Using networks data as an example, it can include data from intrusion detection or prevention systems (IDS/IPS), Firewalls, web servers, and other kinds of devices, including routers and switches, thus the data is comprised millions of events in a short time. The basic security log data is created by alerts or incidents, which generally covers the information of object, behavior, outcome, technique, device group, and timestamp. Containing networks, hardware and applications log data, these datasets must to be distilled to what an analyst can reasonably deal with. The explicit correlation and normalization criterions will bring about effective data analysis.

When a security analyst wanting to see all user logins within a certain time period, have to know what the specific category of each event type is, to retrieve that information. According to a series of taxonomies to extract security events from raw log file, and dividing into different categories, such as authentication, new connection, signature, etc. Some examples are the following (Fig. 14.6):

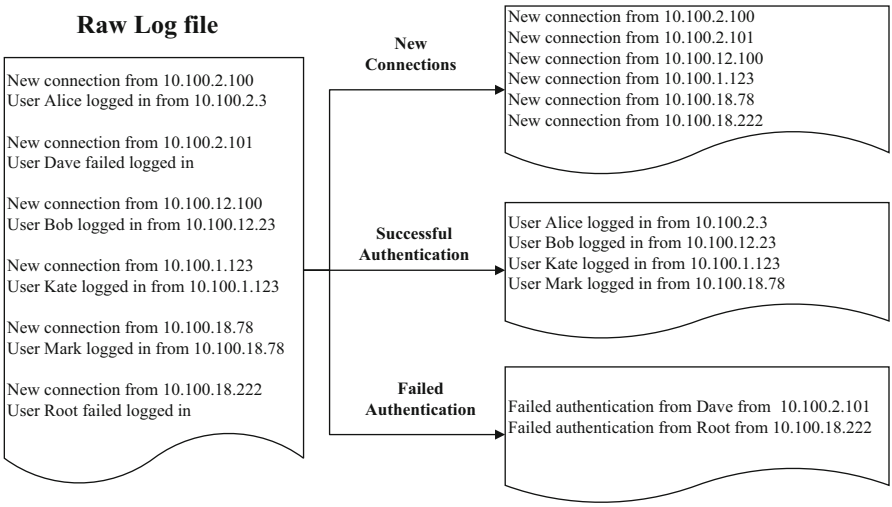


Fig. 14.6 Security events extracted from the logs

14.2.1.1 Criterions of Correlation and Normalization

- Clearly state the policy that lays on what is forbidden and what is permitted
- Translating the severity scales used by the different devices into uniformed standards, such as 1—Very Low, 2—Low, 3—Medium, 4—High, and 5—Very High
- Evaluating the mutual relationships of logs data by category and severity, and how significant that relationship is
- Correlation logs data from all security devices in the cluster base on time, location, and joins across multiple data sources
- Normalizing various formats of the logs data with high performance and integrity
- Creating key metrics to retrieve data according to the policy, this will significant in reducing the number of incidents for analysts.
- Including the workflow that can distinguish any incidents from false positive to a potential attack
- Continually tuning vulnerability assessment and remediation to more precisely and quickly find out when security issues happened
- Recording all the decisions and changes as tuning is iterative
- Creating and configuring index to accelerate data transfer and search

According to the criterions that stated as above, when an organization limits the privilege of 'root' to access a specific configuration of an application in security policy. At the beginning, the data should be retrieved from raw log files of all activities that includes 'root' by key metrics, which defines in policy. And then normalized related data into the unified format to store data into database as the data

source for further analysis. Meanwhile, information of exceptional incidents will be used to enhance and tune the policy and metrics. Tuning key metrics will help as well by making it easy to find out what type of events should be from a device.

Incidents or events can be filter out based on the correlation policy, which requires abundant experience and knowledge on IT security. It is difficult, and the inappropriate correlation will cause severe risks.

14.2.2 Log Data Analysis

Log data analysis determine the meanings and causes of security issues, this is the key part of all efforts in the whole process. While it highly depends on the quality of log file correlation and normalization, whether the event is malicious or not also depends on context of log files. For example, source and destination may become an attacker and target if a network analyzer, such as a HIDS or NIDS, evaluates the traffic as hostile. After distill raw data to structure data, properly analytical tools will present the data in dashboard, report, or pattern discovery and interactive discovery (Fig. 14.7).

14.2.2.1 Criterions of Log Analysis Process

- Analyzing the total number of events—If the event count is higher or lower than normal, seek the key metrics and patterns with the timestamp in log data, and even verify the logging policy inclusive enough.

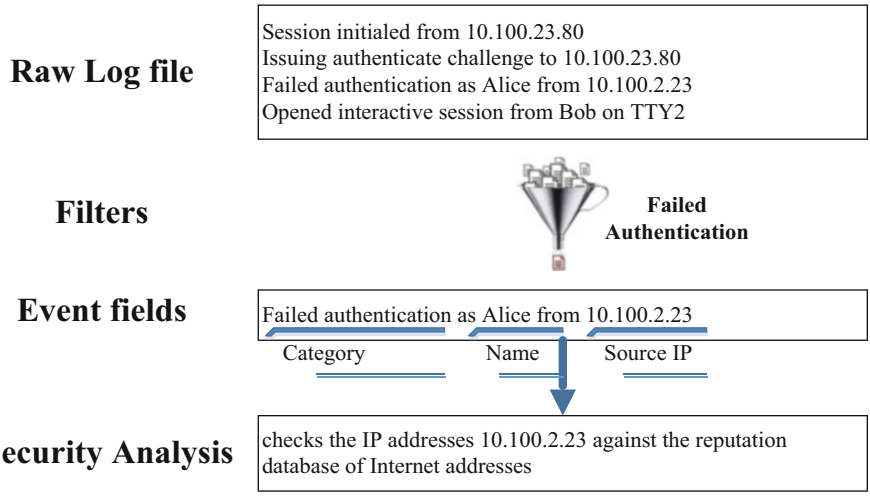


Fig. 14.7 Correlation directives and security analysis

- Analyzing the variety of events—If the number of unique event types is low, verify whether the logging policy includes all the sources, and the key metrics cover various security events.
- Analyzing the number of events occurred periodically—If the number of event count is apparently high or lower than last period, comparing the log data to find out the root causes.
- Analyzing the network behaviors, which is listed but not limited
  - Investigate the source
  - DNS lookup
  - WHOIS lookup
  - Traceroute
  - Show ARP
  - Review network diagrams
  - Contact the asset owner, and discuss what seeing
  - Consider full forensics if it's serious enough
  - Update Anti-Virus signatures and patches, and rescan
  - Reboot the host from a bootable image, and scan the host for malware
- Analyzing the availability, capacity, and performance metrics of hardware—If the trend of the hardware metrics is abnormal, the potential harmful actives may happen or has happened in the system
- Analyzing software security matrices according some compromised methods, which is listed but not limited
  - Injection
  - Authentication and session management
  - Cross-Site scripting
  - Insecure direct object references
  - Security misconfiguration
  - Sensitive data exposure
  - Missing function level access control
  - Cross-Site request forgery
  - Using components with known vulnerabilities
  - Invalidated redirects and forwards

It is very significant that log data from intrusion detection or prevention systems (IDS/IPS), firewall, system and network device logs can be monitored and analyzed in time. And analyzing log data serves several purpose. First, the real-time analysis gives security administrators an overview of what is currently happening on the hostile system and network. Another purpose is to understand how activity changes over time can quickly react to adjust the security configuration if necessary. In additional, the log data analysis result is the most important information to attack forensics.

Several famous open source log analysis tools are released with a general analysis workflow, such as OSSIM, LogRhythm, Nagios, Splunk, etc. In the real environment, open source tools need be customized on log correlation, logging policy, extract filters and index. Analyze all data to quickly reveal trends, spikes and anomalies, and then present the data further with visualization graphs and charts to understand import trends.

14.2.3 Specific Features for SIEM

Previously, we've discussed what SIEM does and how it works. Next, we move on to some special consideration of SIEM. As SIEM provide entire security service for the organization, three specific features—capability, reliability, scalability, should be considered when implementing SIEM.

Capability (Performance) The capabilities of SIEM solution should fit the needs of the enterprise, and provide accurate and meaningful data. Also, the capacity of correlate data among many different sources of data should be considered. Most SIEM systems collect log data in two ways, real time or poll data from the security devices, but each has its own benefits and defects. The real time mechanism requires high performance, it provides data that is very current with minimal latency, while it's unlikely to give inaccurate historical trending analysis. Centralized repository is an alternative method for real time. This mechanism pull data in the repository across the enterprise system, and the data-gathering process is effectively ensure the accuracy of data. However, in the security aspect, real time data is one of the most valuable property.

Computing capacity is a tremendous challenge in the distributed SIEM as well. When data size range into the terabytes, it would not be store in a single location, and any database or storage system decreases in performance. This can obviously cause other serious impacts on the entire SIEM whenever some data are in need immediately but cannot be available. Therefore, by using a hierarchical architecture, the benefits of each methodology can be gained and mitigate some of the disadvantages.

Reliability In the SIEM, all device and system are reliant on a centralized system for configuration, deployment and monitoring of the security policy. A single point of failure will cause a security hole in the SIEM system, and even lead to an unmonitored security environment. Clone a second node for failure recovery can ensure the time to repair the SIEM system is minimized, but in a high cost. Thus, through a regular system backups and test the restores are economic and reliable ways to ensure the system works as expected.

Aside from systematic failures, all of information technology related departments in the organization have the responsibility to maintain the availability of the each security devices. Since the SIEM system had connections to other devices and systems on the network for deployment and monitoring purposes, attackers able to use these connections to proceed further intrusions.

Scalability Scalability, as a primary property of SIEM systems, associates with bandwidth, storage, and distributed computing. Scaling out SIEM architecture helps enterprise to overcome some of the general limitations in information technology systems. It should be considered into the overall SIEM architecture design. No matter what size that enterprise may be, build a scalable SIEM tool to be able to support the business in the future. Certainly, leveraging the expenditure to ensure the implement that can enough meet the current purposes.

14.2.4 Case Study of Log Correlation

Assume that a big company has implemented a logging infrastructure. The company deploys an Intrusion Detection System (IDS) using Snort. Also, the company installs a VPN server, which allows teleworkers to remotely access the company network. The system administrator also maintains an asset database which contains the details of assets in an organization. Examples of asset details are OS, IP address, MAC address, hardware manufacturers, hardware types. Also, Active Directory is used to manage users, authenticating and authorizing users and computers in the network and ensuring only authorized users have access to the company resources. The raw log messages from Snort and VPN server are transmitted to a central storage location using Syslog, shown in Fig. 14.8.

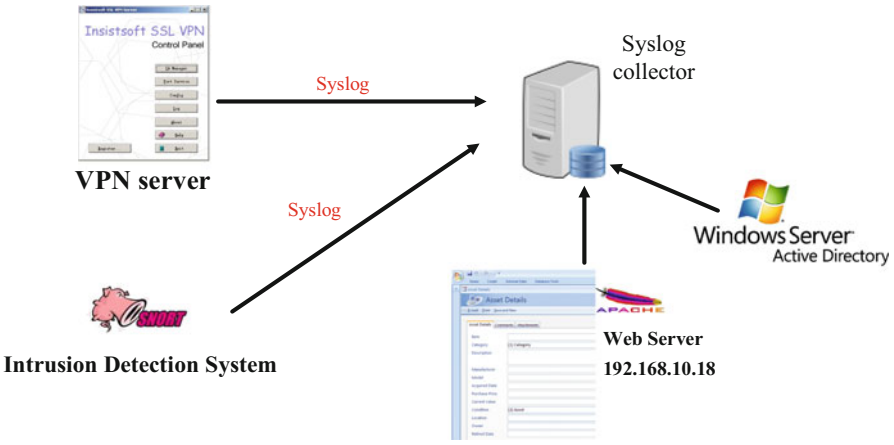


Fig. 14.8 Example logging infrastructure

Suppose that the VPN server records a login event for any given user along with a location and timestamp and forwards it to the centralized log server. Around the same time as the login event, a DoS attack was detected by Snort, originating from 192.168.10.23 against a company computer of IP address of 192.168.10.18. Based on VPN server logs, you find out a user with UID slin is using IP address 192.168.10.23. Through event correlation, we now know that the user with UID slin is DoS attacking a computer of IP address of 192.168.10.18. Further, we search asset database and find out that IP address of 192.168.10.18 is assigned to the company web server, which is an important server for the company. Also, we find the full name of the user with UID slin by querying Active Directory. Assume that user slin's full name is Sheldon Lin. Finally, we can conclude that user Sheldon Lin is DoS attacking the company web server. The conclusion is very helpful in terms of two things compared to the raw log data collected. First, we know who the attacker is. Second, we know this is a critical incident, and therefore it must be given a high priority.

14.3 Implementing SIEM

14.3.1 How OSSIM Works

OSSIM is an open source security event management system and developed by AlienVault, providing security analysts and administrators a view of all the security-related aspects of their system [9]. AlienVault OSSIM comes with OSSEC host-based intrusion detection system, and the architecture is demonstrated in Fig. 14.9.

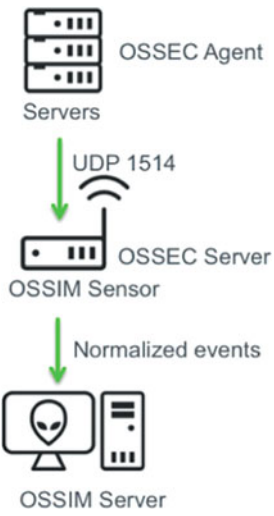


Fig. 14.9 AlienVault OSSIM architecture

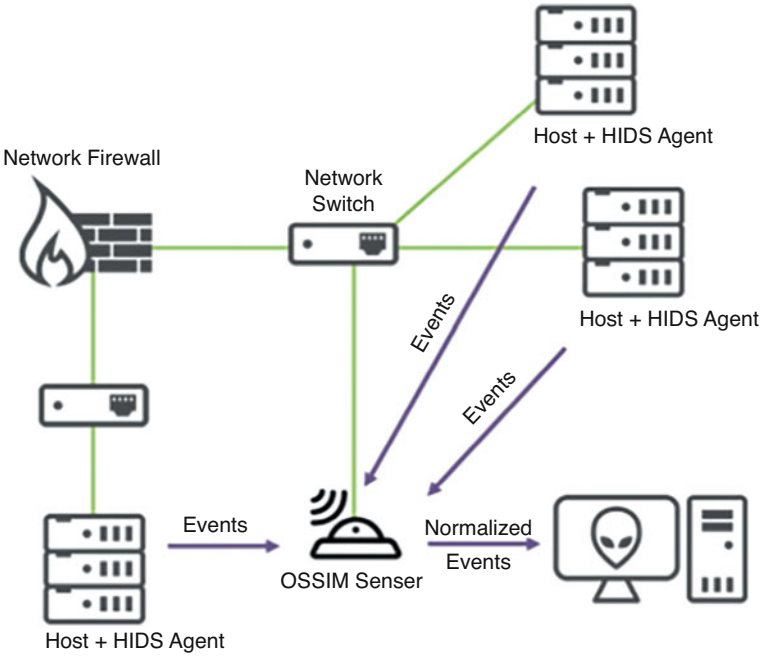


Fig. 14.10 Deploying OSSIM HIDS on hosts

Network interface and assets in the network can be automatically and manually added and configured on OSSIM server. Devices, systems and software that support the Syslog protocol are configured to transmit their log events to the OSSIM sensor over UDP port 514 or TCP port 514. OSSEC Agent running on the OSSIM sensor that is configured with a series of log-parsing plugins, which read the incoming log files.

The HIDS (Host Intrusion Detection System) agent in AlienVault OSSIM looks for suspicious or malicious activity and must be deployed on individual hosts. As scanning and adding new hosts by OSSIM server, HIDS agent can be automatically deployed on the host by OSSIM on UNIX/Linux, Windows or other operating systems. It analyzes operating system log files, looking for changes to system files and software, as well as network connections made by the host (Fig. 14.10).

After HIDS is deployed on hosts, the OSSIM server will going to parse the event priority and reliability. Each event type that has an SID (Security ID) is assigned a priority and reliability score when the plugin is created. The OSSIM server also maintains an inventory of known devices on the network, with an associated asset value to weight against the event's priority and reliability score to produce a risk value (Fig. 14.11):

Fig. 14.11 Parsing events priority and reliability

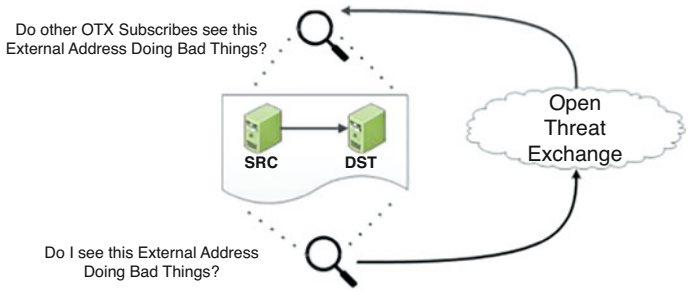


Fig. 14.12 Open threat exchange (OTX) workflow

$$\text{Risk} = \textbf{asset} * (\text{reliability} * \text{priority} / \textbf{25})$$

For correlation and normalization, OSSIM defines a taxonomy of event types that SIDs can be matched to and retrieved. Therefore, correlation directives can correlate events via VlienVault taxonomy allowing the creation of device-independent correlation rules. Different correlation rules may take the same events as input, being able to look for patterns and sequences of events across multiple devices and types.

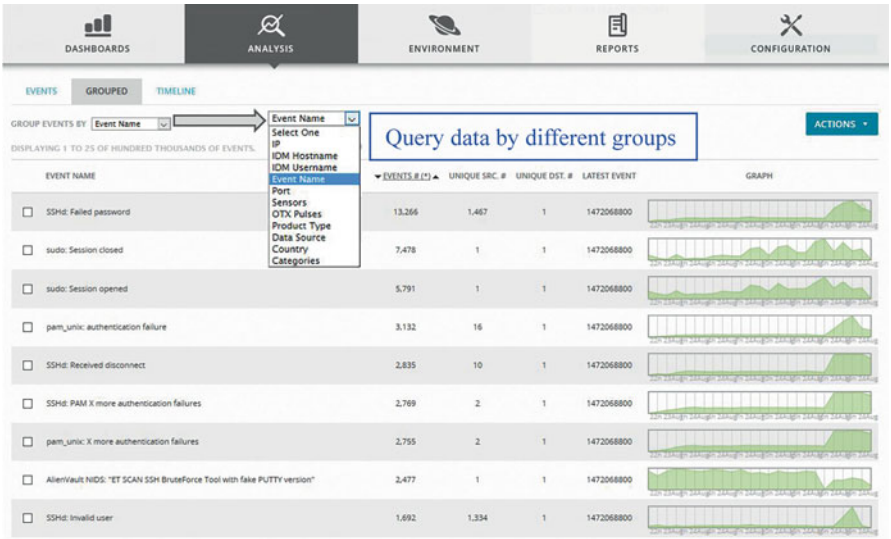
Also, AlienVault establishes an Open Threat Exchange (OTX) for all OSSIM users to crosscheck and corroborate the reputation database. Users can use the OTX database to verify the suspicious information, such as IP addresses. Likewise, Events that indicate attacks from external addresses will be anonymized and submitted back to OTX (Fig. 14.12).

As most of SEIM, events are available for searching and browsing on web UI, and correlation directives alarms can be triggered in certain conditions. OSSIM also provides several types of reports for view and download from the web UI. In next section, several screenshots will show how OSSIM achieves log data visualization.

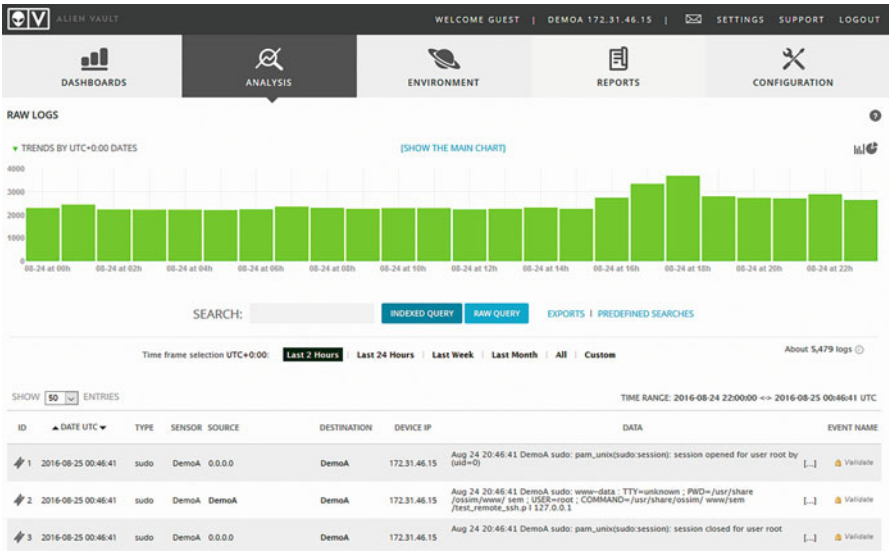
14.3.2 AlienVault Event Visualization

OSSIM provides rich ways to view log data as well as managed assets in different scenarios, which can be seen below.

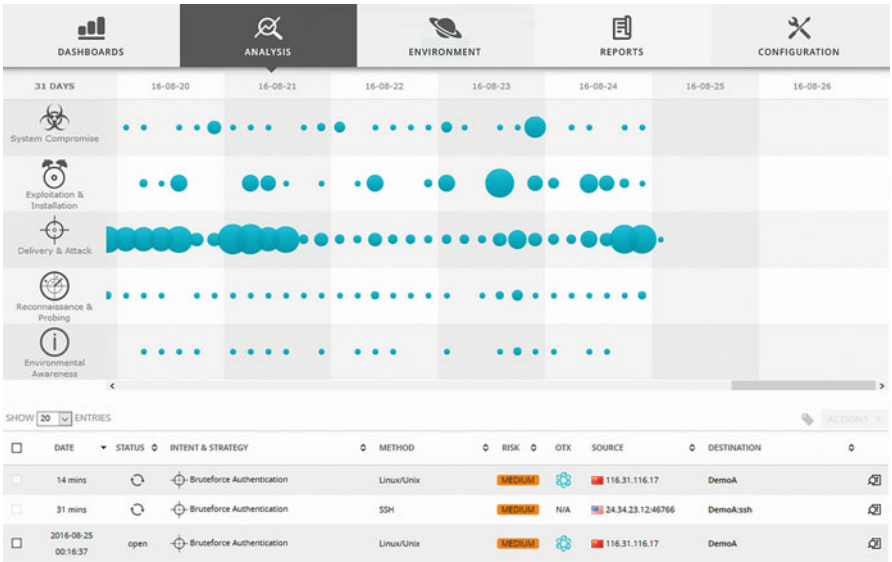
1. **Analysis → Security Events (SIEM)**



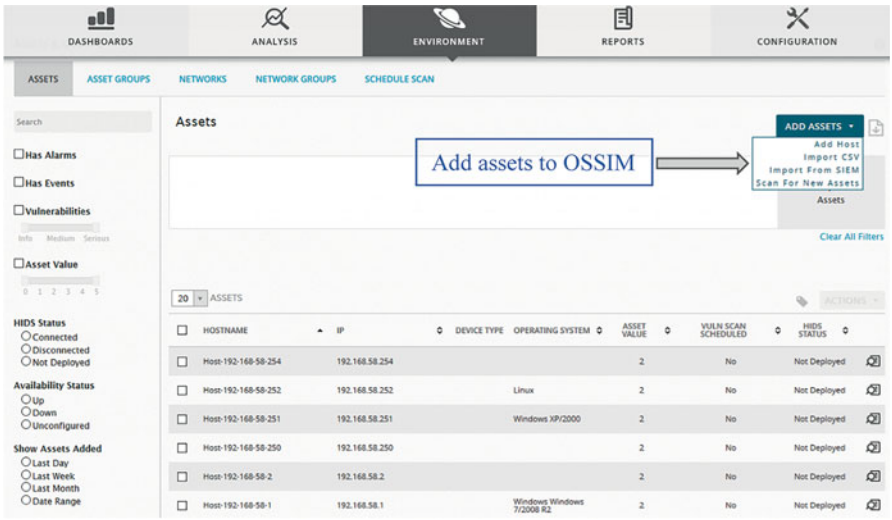
2. **Analysis → Raw Logs**



3. **Analysis → Alarms**

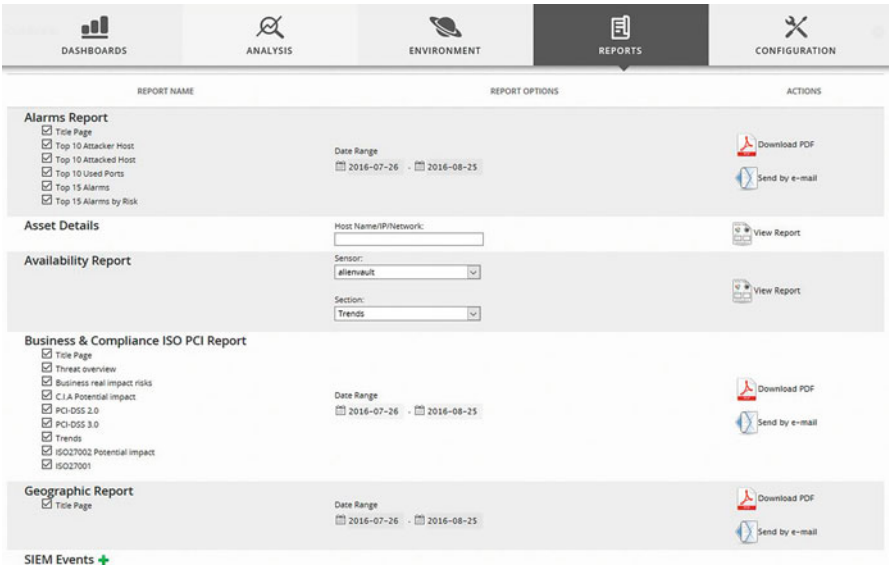


4. **Environment → Assets and Groups**



5. **Reports**

Categories: Alarms, Assets, Compliance, Raw Logs, Security Events, Security Operations, Tickets, User Activity, Custom Reports



Review Questions

1. What does SIEM stand for? What is SIEM?
2. Which of the following is the right format for the selector in the rsyslog configuration file?

   (a) mail,auth.info
   (b) mail;auth.info
   (c) mail.alert,auth.info
   (d) None of the above

3. The destination port number field of the TCP or UDP packet indicates what application protocol is being used. For example, port 22 indicates

   (a) ssh
   (b) http
   (c) smtp
   (d) pop3

4. Which of the following rsyslog.conf line is correct?

   (a) mail,authpriv.info <TAB> /var/log/secure
   (b) mail.info,authpriv.info <TAB> /var/log/secure

(c) mail.*,authpriv.info <TAB> /var/log/secure
(d) None of the above

where TAB stands for Tab Space.

5. Which of the following is NOT found in the rsyslog.conf file?

(a) The program sending the message
(b) how logs will be saved
(c) The severity level of the message
(d) None of the above

14.4 Practice Exercise

The objective of this exercise is to learn how to collect, parse and analyze logs. Specifically, you are required to use the regular expression to develop a log parser which can continuously monitor the log file /var/log/forensics.log below. Note that you can use any programming language with which you are familiar (e.g., PHP, Shell, Java, or C/C++). It should parse out user authentication messages into a standardized format and insert them into a table defined below. Finally, you will practice log analytical skills.

14.4.1 Setting Up the Exercise Environment

In this exercise, you will use MySQL to store user authentication events including successful logon/logoff and failed login attempts in Kali Linux. MySQL is installed and configured by default in Kali. Note that by default there is no password for superuser "root" for MySQL server. Next, you need to create a new database called "forensicsdb" and a table named "event", which is used to store user logon/logoff activities in a standard format. Below is an example of the schema of the "event" table, which can used in this exercise.

Field (Column name)	Data type	Description
Event_id	INT	A numerical primary key value which will automatically increase whenever a record is inserted into the "event" table
type	varchar(24)	Event classification, such as user authentication such as user log-in and log-out. Examples include auth.login.success auth.login.failed auth.logoff
username	varchar(24)	The name of the user

(continued)

Field (Column name)	Data type	Description
s_ip	INT	The IP address of the machine where the user is connecting from
s_port	INT	The source TCP port number of a connected session, or in our example, it is null for login at the console
d_ip	INT	The server's IP address. In our example, it is the IP address of Forensics Workstation (or Kali Linux VM)
d_port	INT	The destination TCP port number of a connected session. In our example, it is TCP 22 for SSH or null for login at the console
time	datetime	The time when the session is established

- Start/Shutdown MySQL Server (mysqld)

MySQL is run as a service called "mysql" (configured at "/etc/init.d/mysql"). However, it is not started automatically in Kali after boot. To manage mysql server, you could open terminal and type the following commands:

```
// Show the status
$ sudo service mysql status
// Stop the MySQL Database Server
$ sudo service mysql stop
// Start the MySQL Database Server
$ sudo service mysql start
// Restart the MySQL Database Server
$ sudo service mysql restart
```

Note that MySQL is a client-server system. The database server runs as a server application. There can be many client programs but there can be only one database server, and these clients communicate with the server; that is, they query data, save changes, etc. Users can access the database server via a client program, locally or remotely thru the network, as illustrated (Fig. 14.13):

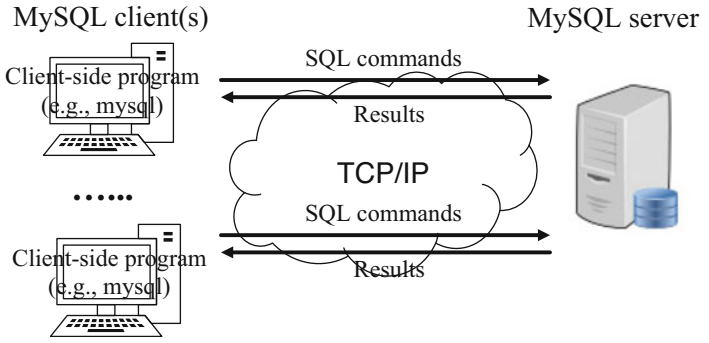


Fig. 14.13 How MySQL works [10]

1. Start/Stop MySQL Command-line Client (mysql)

   (a) Start a client as superuser "root" (-u), and prompt for password (-p)

   mysql -u root -p

   Press Enter when prompted for password. Recall that there is no password for the root for MYSQL server in Kali Linux.

   (b) Create a database called "forensicsdb"

   MariaDB [(none)]> create database if not exists forensicsdb;
   Query OK, 1 row affected (0.00 sec)

   (c) Create a table called "event" in the database "forensicsdb".

   Use "studentdb" database as the default database

   MariaDB [(none)]> use forensicsdb;
   Database changed

   Create a new table called "event" in the default database "forensicsdb"

```
MariaDB [forensicsdb]> CREATE TABLE IF NOT EXISTS event (
-> event_id INT(11) NOT NULL AUTO_INCREMENT,
-> type VARCHAR(24) DEFAULT NULL,
-> username VARCHAR(24) DEFAULT NULL,
-> s_ip INT(4) UNSIGNED DEFAULT NULL,
-> s_port INT(4) UNSIGNED DEFAULT NULL,
-> d_ip INT(4) UNSIGNED DEFAULT NULL,
-> d_port INT(4) UNSIGNED DEFAULT NULL,
-> time DATETIME DEFAULT NULL,
-> PRIMARY KEY (event_id)
-> );
```

   Query OK, 0 rows affected (0.01 sec)

Once it is completed, the table named "event" is ready to store users logon/logoff activities in a standard format. For example, insert a new record into the MySQL "event" table by using the following mysql statement

```
MariaDB [forensicsdb]> INSERT INTO event (type, username, s_ip, s_port, d_ip,
   d_ port, time)
   -> VALUES ('auth-login.success', 'root', INET_ATON("192.168.44.136"), 8080,
      INET_ATON("192.168.44.136"), 22,STR_TO_DATE('12-01-2014
      00:00:00','%m-%d-%Y %H:%i:%s'));
Query OK, 1 row affected (0.01 s)
```

   (d) Exit MySQL command prompt

```
MariaDB [forensicsdb]> quit
```

14.4.2 Exercises

Part A: Configure Syslog
In this part of the exercises, you configure the rsyslog facility on your Forensics Workstation (or Kali Linux VM) to track user authentication activities, such as logons, and send the logs to /var/log/forensics.log. Then, you generate some logs by logging into Forensics Workstation with both correct and wrong passwords.

Part B: Develop a Log Parser
In this part of the exercises, you are required to develop a log parser, which is able to parse out some important information about user authentication activities. Specially, your parser should at least be able to do the followings:

- Continuously monitor the log file /var/log/forensics.log
- Parse out user authentication messages into a standardized format and insert them into the table "event" created above
- Convert timestamps in raw log data to UTC (Coordinated Universal Time, formerly Greenwich Mean Time(GMT)).

Part C: Log Analysis
In this part of the exercises, you are required to develop SQL search queries to answer the following questions

Q1. Who logged into Forensics Workstation last night between 9pm and 11pm?
Q2. How many failed login attempts since the last successful login of user root?
Q3. When is the last login time for user root?

References

1. Basics of Forensics Log Analysis. https://www.paladion.net/blogs/basics-of-forensics-log-analysis
2. D. V. Forte, The "Art" of log correlation: Tools and Techniques for Correlating Events and Log Files. Computer Fraud & Security, Vol. 2004, No. 8, pp. 15–17, August 2004.
3. Event Correlation across Log Files: What is it and Why is it Important? https://www.accenture.com/us-en/blogs/blogs-event-correlation-across-log-files-what-is-it-and-why-is-it-important
4. N. M. Ibrahim, A. Al-Nemrat, H. Jahankhani, R. Bashroush. Sufficiency of Windows Event log as Evidence in Digital Forensics. Proceedings of the 7th International Conference on Global Security, Safety & Sustainability (ICGS3). Greece, August 2011.
5. The Syslog Protocol. https://tools.ietf.org/html/rfc5424
6. https://syslog-ng.com/
7. https://www.rsyslog.com/
8. How to set up Syslog-ng server on Debian. http://oscarhjelm.com/blag/2013/02/how-to-set-up-syslog-ng-server-on-debian/
9. https://www.alienvault.com/products/ossim
10. http://www3.ntu.edu.sg/home/ehchua/programming/sql/mysql_howto.html

11. Seyed Morteza Zeinali. Analysis of security information and event management (siem) evasion and detection methods. Master Thesis, Tallinn University of Technology, 2016
12. Security Enhanced Linux (SELinux). https://github.com/SELinuxProject
13. https://www.accenture.com/us-en/blogs/blogs-event-correlation-across-log-files-what-is-it-and-why-is-it-important
14. Network Intelligence Corporation. http://www.network-intelligence.com