Windows Event Log Visualization and Query

**CHAPTER 1**

# **INTRODUCTION**

Windows Event Logs, these are logs which has been developed by the Windows operating system to abduct notable activities and events. These logs hold key information about security-related events, applications, and system processes. Each log entry contains information and data like the source, event ID, time and date, and a narration of the event. Event Logs are vital for troubleshooting, monitoring system performance, and locating security issues or unauthorized access attempts. Administrators can use tools like Event Viewer to analyze and access these logs, helping them gain insights into identify potential issues and the system's health. Proper management of Event Logs, that includes regular monitoring, archiving and maintenance, that is essential for ensuring the stability, smooth operation of Windows-based systems and security.

These window event logs also enroll computer’s notification and alerts, Microsoft states about event as any type of occurrence in the system or device which requires users to be alerted to entry added to a log. Or furthermore Linux supports a lot of system logs that helps in administrating the Linux system. file */var/log/messages,* is the most important log, that records a wide rage of events, that includes system startup, system error message and system shutdowns.

[6] It makes categories of each event with the severity level that orders as information, verbose, warning, error as well as critical. The other tools to view window event logs are Sumo Logic log management and log analytics, Site24X7, Datadog, and SolarWinds Security Event Manager (SEM).

[7] The location of window event log is C:\WINDOWS\system32\config\ **folder**. Users can check event logs with the help of 'Event Viewer' to keep track of troubleshooting in the system. The process to check is.

* Now press the **Windows key + R** on your keyboard, for the opening of the run window.
* Type in **eventvwr** and click OK.
* Expand the **Windows Logs** menu.
* Under the **Windows Logs** menu, you will notice different types of event logs—application, security, setup, system, and forwarded events.

**Event log entry**

Each of every event suppress the following information:

* Time: the time when the event took place.
* Computer: name of the device.
* Date: the date whenever the event took place.
* User: the username of the user whenever the event took place.
* Type: type of event like warning, error, information, security failure audit, security success audit.
* Event ID: a window identification number that designates the event type.
* Source: the program or feature that caused the event.

**Types of event messages**

The categories are:

1. Information event: it states the successful completion of the task, for example the installation of an app.
2. Error message: states the notable problem that causes in the loss of functionality.
3. Warning event: notifies the authority of a possible problem, for example low disk space.
4. Failure audit: it states the failure of audited security event, for example when a user ended up locking himself out by entering wrong password.
5. Success audit: it shows the completion of audited security event, for example logging in successfully.

## **Windows Log event Categories**

The further categories of event logs are:

1. Application event log
2. System event log
3. Security event log
4. Setup log
5. Application and service log
6. **Application event logs**

The application event log [9], it is one of the most accepted event logs in the Windows operating system. It is an element of the Windows Event Logs; this element enrolls events that relate to programs and applications running on the system. When applications come across warnings, crashes, errors, or other notable events, they can develop entries in the application event log. Some general types of events enrolled in the application event log include:

* Application errors: This type of event happens when an application experiences an exception and error that causes it to crash or malfunction.
* Application warnings: Warnings can be logged, whenever an application encounters a non-fatal condition or issue that might crave attention.
* Informational events: App can log informational events to administrates data and information about their successful processes, normal operation, or other information which is relevant.

The application event log is an irreplaceable resource for system developers and administrators, as it assists them diagnose and identify issues with specific applications already installed on the system. Administrators can troubleshoot application-related problems as well as ensure smooth functioning of software on the Windows platform through analyzing the events in the system log.

1. **System event log**

The System Event Log [10], it is one of the ordinary event logs in the Windows operating system. System event log is accountable for record-keeping events that are related to the Windows operating system services and elements, furthermore it is one of the main components. The registered events in the System Event Log enables us vital information and data related to the system's hardware, functioning, and events that are related to driver, and as well as other system-level activities.

There are some standard types of events that were recorded in the System Event Log that includes:

* Device driver installations and failures.
* Service starts and stops events.
* Hardware errors or malfunctions.
* System critical errors and warnings.
* System startup and stop events.

The System Event Log is significant for system support panel because it guides them to track wholesomeness and regularity of the window operating system and furthermore administrates them. Executives will be able to troubleshoot driver-related and hardware issues, analyze the problems that can affect the entire system, and identify system-level errors through analyzing the recorded events in the log.

**Device driver installation and failure**

This is a software program which enables communication between operating systems and hardware devices. Hardware devices which include graphic cards, printers, and network adapters. An event is recorded in the system event log whenever a device driver is updated or installed. This event generally notifies about the error occurred in installation or whether it is getting successful. In case of failure, it leads to system instability, hardware’s malfunctioning, and further compatibility issues.

**Service starts and stops event.**

It is a background process that enables computing and running on the system. Start and stop events will be recorded in this system event log. They inform executives about the position of critical system services and their accessibility. This element is important for analyzing if the services are getting perfectly generated or not, as well as analyzing possible problem with the service dependencies.

**Hardware error and malfunction**

The system event log records the event whenever any element meets the malfunction or error. Some of the errors relates to hard drives failure, hardware related issue, RAM module malfunctioning, and overheating of processors. To analyzing faulty elements and then proceeds mind in one manner by repairing or replacing the hardware, all you need is to monitor to keep an eye on these errors.

**System critical errors and warning**

It indicates the possible problems and serious issues which require attention. Events like system crashing, critical issues which can affects the performance of the system and Kernel-level errors. Addressing these errors is important to maintain the health of the operating system.

**System startup and shutdown events**

These events are registered in the system event logs, that includes the details about the starting and stop times of the system as well as timestamps. These events are a handful for understanding system uptime and further issues which can occur while processing.

1. **Security event log**

This is a common event log in the window operating system and a main component of window event log which is responsible for the documenting of activities and security-related events. These events enable us to provide key information related to access control, user authentication, security-related activities, and changes in security policies.

Types of events which can be recorded in the security event logs are:

1. Object access: gain pursuit for accessing sensitive objects and resources on the system, for example folders, registry keys and files.
2. Audit policy changes: locate the changes in the security audit policies on the system.
3. Successful and failed logon attempts: it records events whenever an unauthorized login attempt fails and if the user log-in successfully.
4. Security group membership changes: it records the events during the modification of security group membership.
5. Account management: it records the changes whenever any user changes the password, while deleting the account and creating the account.

This event log is important for the maintenance and monitoring of the security of the window operating system, which helps system security when detecting any dubious activity, while helping the executives, examining security incidents, and identifying possible security breaches.

1. **Setup log**

It records the events while setup of software, installation, and device drivers on a window operating system. Window logs the installation procedure in the setup log whenever updating and installing the new software. It enables the important information for monitoring the installing of the process and troubleshooting and as well as it locates the outcome of the generating the software and traces the steps. This setup log is really a handful while facing the issues in driver updates and installing the software. The log will gather the details about description, details which is relevant, error codes and the files which is involved, in case if any problem occur in the setup.

User can access it by window event viewer, maneuver to "Applications and Services Logs" > "Microsoft" > "Windows" > "Setup," to find the setup events.

It will help to assist in troubleshooting while installing drivers and installing any program in the windows system.

1. **Application and service log**

It includes the custom event logs generated by services that are already installed in the window operating system and by wide range of applications. Third-party services and programs may generate the logs under application and service logs, non-identical or far from the standard event logs which are generated by the window itself. It enables a vision to operators to record information related to the service and application. These logs are a handful for diagnostics purposes, monitoring and troubleshooting, because they allow more control over the information and events they want to be achieved.

For accessing these logs, we need to follow these steps:

* Press window key + R to open the event viewer, type “eventvwr.msc” and hit Enter.
* Maneuver to “application and service log” on the left pane.
* Users can be able to see a list of custom logs which are created by different application and services. In the main pane on the right side, view the recorded event log.

They provide the important insights into activities of the service and application, and their behavior.

**CHAPTER 2**

# **Literature review:**

Forensic investigations involve log investigation because it tends to make investigation relatively easy and accelerates security. Logs provide important information about the intruder activities which could be involved in hacking of systems and website, DOS attack. Audit logs are very complicated log events and format is in binary form. Moreover, windows log events are decentralized in nature. To cope with these complexities and emphasize the importance over In this study [1], The author recommends a reasonable alternative for decentralized log storage which then seems to be using Win Syslog which is a central Syslog server and event reporter that can translate the data into syslog format from the binary format.

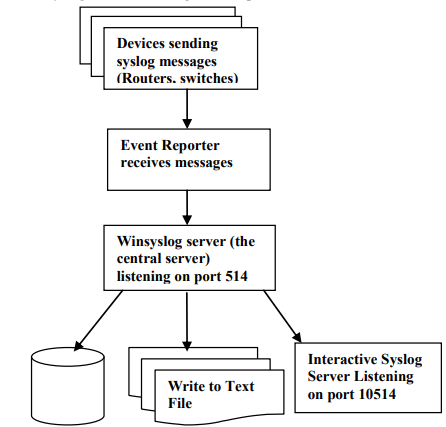


Figure 1: Different sources of logs collecting in Winsyslog [1]

Further study concludes that [2], Cybercrime is growing ever more relatively common, thereby further it needs to be evidence already when individuals may very well be accused of violating the law. Scientific proof, legal requirements, and Windows OS source materials as well as the Tracking system, slack space, and event log seem to be the major talking points of this research project. A VMware computer system was utilized to simulate cybercrime operational processes the same as computer-controlled password guessing and getting hacked on such a Windows operating system. Event information and data have been thoroughly investigated for weight and social acceptance by windows log investigation.

In this study the author concludes that [3], the Windows event log is a flawed tool used in digital forensic cases, often not being considered verifiable. This dissertation aims to develop a solution to address flaws in the Windows event logging service. Research found that disabling the service allowed for modification of important data and easy transplantation of event logs. A C# and Microsoft.NET-based event logging application was developed, using RSA and AES encryption and HMAC hash signatures to improve data integrity. The application consists of an event logger, an event viewer, and a symmetric and asymmetric encryption. The symmetric encryption was found to be 800% faster than asymmetric encryption, and the HMAC hash signatures were tested for brute force attacks.

In this approach [4], A Windows event forensic entire process is explained in this article (Win EFP) to try and understand the event log files with the Windows operating system. The Win EFP deals with forensics of logs of activities and occurrences that take place in Windows. As a consequence, it aims to provide guidance to forensic investigators on continuing to investigate digital forensics via Windows event logs. Including both business and home environments, Windows happens to be the most widely utilized operating system. The Windows event forensic service expects that the event logging service is not clear and unambiguously ended up turning off by a Windows user. Win EFP could be employed in almost any forensic official investigation. that included a computer windows XP. Many Windows event log items that used to be extremely important for forensics were catalogued and identified in process of WinEFP development.

The proposed theory states that [5] logs contain a great deal of runtime information; logs are essential to the design and administration of software development system. As the complexity and size of software are increasing, the volume of logs are also increasing. As there is a shortage of openly accessible data sets and standardization, so only a small proportion of AI-powered log analytics techniques have already been successfully implemented in the sector over the past several years. Log hub had also managed to gather 17 real-world log datasets from those of a wide range of technologies, which include supercomputers, distributed systems, mobile systems, operating systems, server applications, and dedicated software. The paper also provides a situational analysis on intrusion detection system, describes the data, and introduces real-world usage specific cases. Including around 77 GB of data, Log hub is a 17-log dataset enabling log analytics that uses artificial intelligence. It supports the log activities of the organization and continues to act as a discussion board for performance monitoring, datasets, and comments from business and the academic community. Log hub has indeed been downloaded by over 380 organizations, and the platform is planning on keeping trying to gather clearly labelled datasets and planning to release open-source log analysis toolkits.

## **Technology and tools**

1. **Window event viewer for log visualization and query**

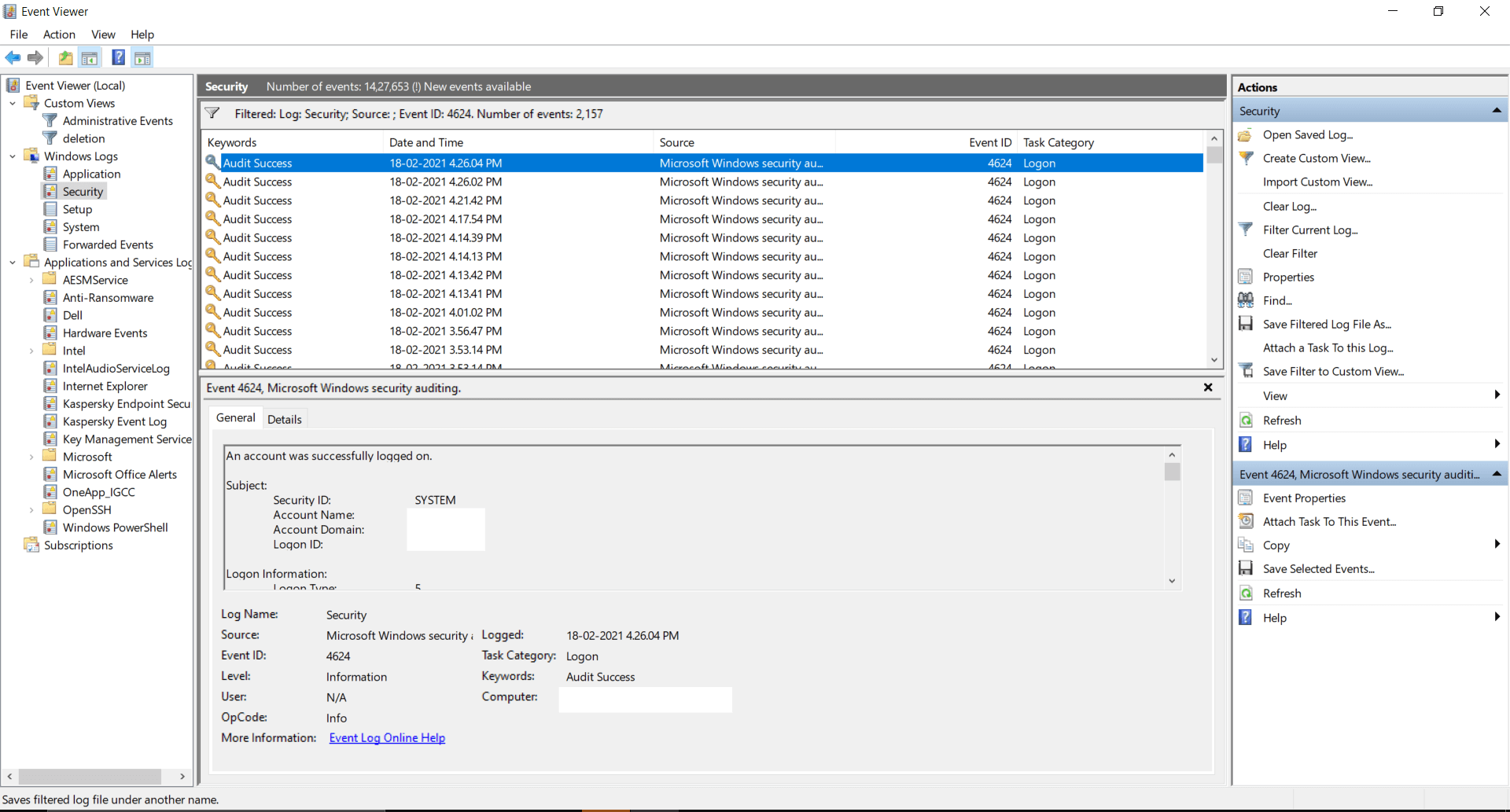
The Windows Event Viewer is a dynamic tool which administer query log events and visualization for event logs in the Windows operating system. Furthermore, it also grants users the opportunity to analyze and view numerous event logs, as well as authorize advanced querying to refine and search for events that are based on numerous criteria.

To employ the Windows Event Viewer for log visualization and querying:

* Firstly, open Event Viewer: Press the Windows key + R, for the opening of the Run dialog box, type "EVENTVWR.MSC", and press Enter. This will open the Event Viewer application.
* operate to Event Logs: In the Event Viewer window, you will be able to find wide range of event log folders on the left-hand side, that includes security, application, System, and others. Click on the significant log folder to view its recorded events.
* View Events: Clicking on a log folder will display the recorded events in the main pane on the right. You can scroll through the events to examine their details.
* Filter Events: To perform a basic filter, click "Filter Current Log" in the Actions pane on the right. You will be able to filter events by different parameters like keywords, event IDs, event source, as well as date ranges.
* Create Custom Queries: For more up-to-date querying, use the "Create Custom View" option in the Actions pane. This will grant you the ability to build complex queries that are based on specific criteria, such as keywords, event IDs, event sources, event levels, and others.
* Save Queries: user can save custom queries for the future use and easy access.

Through employing the query capabilities and log visualization of the Windows Event Viewer, operators will be efficiently analyzing event logs, monitor system performance, troubleshoot issues, further investigate security incidents, ultimately helping to maintain the security and health of the Windows environment.

[8] it is necessary to have “security information and event management” (SIEM), this solution not only aggregate the logs rather they also provide user to efficiently visualize the data and gather evaluative perception. The old method of analyzing the logs is:



**Features of event log analyzer**

* SEIM
* IT compliance management
* Reporting console
* Log management
* File integrity monitoring
* Log forensic analysis.

1. **NirSoft utilities for log visualization and query**

NirSoft provides several tools for system troubleshooting, security, and maintenance. They would not provide a specific tool allocated entirely to querying and log visualization like Windows Event Viewer**.** Nevertheless, NirSoft's assemblage of services includes diverse tools that can be helpful for accessing and analyzing defined types of logs on Windows systems. For example:

* Nir-Soft's "Event Log Sources View" service allows the user to view a list of all event log sources equipped on your system, that also provides data and information related to each source as well as their associated event logs.
* "My Event Viewer" is another Nir-Soft tool. This tool presents a more user-friendly terminal than the default Event Viewer, authorizing the user to view event logs in an uncultured manner.
* "Last Activity View" presents an inclusive report on various system activities, such as user login, system shutdowns, and application installations. which can be handful for monitoring system troubleshooting and usage.

On other hand, While NirSoft activities offer beneficial performance for system analysis, they might not be able to cover the entire range of characteristics enabled by the local Windows Event Viewer. In case the user needs more up-to-date log visualization capabilities and querying, it is better to count on the local Event Viewer tool or investigate other functional log analysis tools that are available in the market.

1. **Splunk for windows log visualization and query**

Splunk is the most powerful log management system and analysis platform and an internationally well-known system, which can be utilized for querying and log visualization and includes window log data. It proposes indexing, application logs, hefty abilities for ingesting, system logs, examining log data from different sources, security logs, and window event logs and more.

* **Advanced querying**: Splunk’s search processing language (SPL), offers the user the chance to search on the log data and perform complex queries. Users can correspond and filtrate, and analyze the time range, event types and as well as keywords.
* **Data visualization:** Splunk enables a user-friendly affiliate to generate interactive visualization and dashboards. Through log data, users can generate tables, maps, graphs, and charts to gather insights.
* **Indexing and parsing**: Splunk indexes and parses the log data to make it easy to search for the user as well as enabling efficient and fast querying after ingestion.
* **Data ingestion:** By using different methods like window event log monitoring, Splunk universal forwarder, or HTTP event collector, Splunk allows the users to ingest the data from the window machines.
* **Reporting and analytics:** to initialize the analytic based on log data and custom reports, Splunk proposes the reporting characteristic. These can be scheduled for distribution and automatic generation.
* **Alerting and monitoring:** based on the custom searches and already defined conditions, Splunk allows users to set up notification and alerts.

**Chapter 3:**

* **windows log event Data Collection and Extraction [1]**

Event log entries in Windows systems are generated and obtained as components of the Windows Event Log collection, analysis, and processing conditions for management, observation, troubleshooting, and multifactor authentication.

The steps in this process are organized as follows:

**1. Event Log Types and Channels:**

Windows Event Logs are split into different groups, each of which represents general and especially channel-specific attributes of security events. Although there are three basic categories:

* **Application:** Framework logs and Operating system.
* **Security:** Document security-related incidents include login attempts that appear to be trying to authenticate legislative alterations and human rights abuses.
* **System:** Manages to capture startup and shutdown events, driver limitations and hardware, and network configuration.

**2. Identifying Relevant Channels:**

Must use the event log channels that have been influential in their intellectual aims and observation. Choose another channel that directly relates to their key targets because they're all going to accumulate a broad spectrum of activities.

**3. Accessing Event Viewer**

Windows' Event Viewer program offers a GUI for retrieving information and representing event log goods and services. It's either a platform one that's helpful for detailed analysis or that's already readily accessible.

**4. Using PowerShell for Extraction:**

The procedure for gathering information can be accomplished and organized via PowerShell test scripts. Like using 'Get-WINEVENTT' and perhaps other PowerShell cmdlets to remove unwanted event usernames and passwords. For example, in the article:

Get-WinEvent "Application" -LogName (Get-Date).

AddDays(-7)

**5. Filtering Events:**

Events may very well be categorized employing PowerShell; obviously, it depends on several different parameters, which would include the designated period, event ID, keywords, and sources. The personal information users retrieve is diminished through detection to include only valuable data.

**6. Exporting Data:**

Event log data that has been managed to gather and allowed to settle may be produced and sold in a broad range of formats, which would include XML, CSV, and even a personalized database. In-depth interrogation has now become entirely feasible using advanced and powerful software or technological devices.

**7. Using Third-Party Tools:**

There are third-party alternatives that are already considered experts at gathering and evaluating event log data from various websites throughout the whole network, including SIEM (Security Information and Event Management) systems.

**8. Remote Event Log Collection:**

With a portable event log personal library, users might very well gather up event data from some of the other computer systems connected to the identical network, as long as ownership permits. The portable functionality of PowerShell allows this to be accomplished.  
**9. Automated Collection and Scheduled Tasks:**

With Task Scheduler, PowerShell practices may very well be originally planned to be organized at predetermined time intervals. By continuing to support commonly performed fully automated intelligence gathering, you can ensure users have a consistent stream of event information that can be analyzed.

**10. Data Analysis and Visualization:**

Until the event log data is assessed, it may very well be explored using a variety of instruments that range from simple and effective scripts to artificial intelligence and machine learning systems. Data interpretation may be aided by analytical techniques such as Power BI, Splunk, or ELK Stack.

* **Data Preprocessing**
* **Data Parsing:**

Data parsing is the manner of subdividing extremely complex and poorly structured log data, including event log entries, in and out of surrounding nature or character traits that have been simple to handle, draw conclusions from, and process. Raw log data more and more often manages to combine chronological data alongside event IDs, event sources, characterizations, and some other relevant information. In the interests of creating and implementing the ideas of the log entry, data parsing essentially involves indexing and trying to extract certain sources. Data parsing in either the frame of reference of the paper might also necessarily imply:

1. **Log Entry Segmentation:**

Demographic segmentation of log entries into several more different components, such as sequence numbers, event sources, event IDs, and event descriptions, is the first step.

1. **Field Extraction:**

Attempting to create relational database variables by removing the unwanted unique characteristics from either the log or the individual product Attempting to take the timestamp, user ID, and resource that would have been allowed access from a security event log entry, for example, in this case.

1. **Contextual Information:**

Summarizing additional semantic information, such as IP addresses, machine names, or user roles, that places particular emphasis on the frame of reference of something like the event

1. **Handling Variability:**

Trying to handle points of difference in log formats because they might appear to come from many sources of information and systems and seem to have different characteristics.

* **Data Normalization:**

The procedure known as data normalization attempts to put the interpreted log data into something like a reliable, consistent, and symmetric version. By guaranteeing that all of the information is racially homogenous, this strategy reduces the effort needed to observe differences, and consumers are attracted to the numerous different log entries. Data normalization throughout the general framework of the paper would have to include:

1. **Timestamp Alignment:** Working to create a universal timestamp configuration from several different formats with the purpose of providing consistent quality for time-based characterization.
2. **Standardization of Event IDs and Sources:** To make the distinction and assessment of Events relatively easy, ensure consistent Event IDs as well as Sources.
3. **Data Unit Conversion:** For concentration to be measured, transmit data units (which include bytes, or rather kilobytes) into something like a dependable unit.
4. **Value Mapping:** To reduce manual analysis and diminish complexity, measurements are segmented into a previously agreed collection of subcategories or bar codes.
5. **Managing Missing Data:** Had used unsupported assertions and perhaps other suited sorting to come to grips with insufficient or absent data sets.
6. **Encoding Categorical Data:** Attempting to create graphical representations of discrete variables (including certain produced content and user roles) that computers can process.
7. **Data Transformation:** Working to improve the statistical distribution of information through the implementation of conceptual modifications (such as logarithmic transformations)

To construct an adequate dataset that can be used as a foundation for further observation and optimization techniques, comprehensive data manufacturing and normalization are extremely important [2].

* **Data Filtering and Deduplication:**

The publications will more assuredly help in providing cutting-edge techniques or instrumentation for querying the database and data compression for event log administrators, including the setting of their investigation [3]. These technologies contribute to enhancing the caliber of the data that can be utilized for analysis, or something that contributes to the effectiveness of observing analysis and planning treatment plans. The "Event Log Filter" network, which would also be implied in the introduction, could very well incorporate various stages or individual components that work collaboratively to provide good quality and sensor information in an effective manner during immediate and almost specific circumstances.

* **Data Filtering:**

The organized method by which to select specific desired log entries that satisfy specific criteria when resisting others is commonly referred to as data filtering. This technique allows for the elimination of unimportant, redundant, duplicated, and irrelevant features, focusing the analysis exclusively on directly relevant and the most important occurrences. In the historical context of the research, the words Event Log Filter likely represent a program or a power structure that employs advanced and complex techniques and algorithms on event log information to successfully purify and organize the information for further investigation. The manuscript could cover an extensive and broad range of information filtering aspects, like:

1. **Criteria Selection:** Establishing the standards on the core principle of which events are to be incorporated or effectively removed Filtering by data, event source, perceived severity, time window, and so many other standards may have an impact on this.
2. **Redundancy Removal:** Attempting to remove irrelevant information that would save cutting and pasting diagnostic initiatives.
3. **Noise Reduction:** Focusing on identifying and limiting progressively explicit or completely meaningless events that sometimes adversely influence explanatory survey results or cause incorrect strong points.
4. **Online filtering:** Promoting filtering in everyday or almost real-time to reduce computation flexibility interruptions and activate a comprehensive response to future events or situations.
5. **Data Transformation:** The method used to transform raw data event log information into an organized form that encourages individual filtering based on an array of minimum standards.

* **Deduplication:**

The mechanism of deduplicating tends to involve efforts to delete and find unused event log entries from something like a dataset. Network problems, system bugs, and many other contributing factors could perhaps cause duplicate events, which can sometimes lead to research results, incorrect analysis, and budget overruns. the above implementation of traditional topics might be comprehensively addressed in the research article:

1. **Duplicate Detection:** An approach for trying to locate event data that has been exactly equal or way too like the original data.
2. **Scalability:** Working to make sure deduplication security measures are valuable and cost effective, and even more so in present conditions with large event log archives.
3. **timestamp consideration:** Failure to manage duplicates that have the exact timestamp, especially when managing frequently occurring events, seems to be another component to keep in mind.
4. **Duplicate removal:** The techniques of reducing the number of duplicates when managing to keep the data and information.

* **Centralized Log Management:**

To improve the efficacy of data presentation, log information gathering, and problem resolution, a central management organization describes the methods of stockpiling, information accumulation, and summarizing log information from numerous sources in a centralized location. The ELK Stack (Elasticsearch, Logstash, and Kibana) is utilized in the present study [4] to either provide cost-effective centralized log management or by providing a complete representation of the software's functioning and state of health, centralized log management should make troubleshooting simple and easy. Decision-makers are capable of immediately narrowing down reasonable explanations for problems and responding in a timely manner.

Considerable amounts of log information gathered from various places might also be accommodated using central management storage advancement alternatives like the ELK Stack without removal.

The article basically outlines why centralized log management is executed through ELK Stack, more so than Elasticsearch, Logstash, and Kibana. This would try to explain methodologies for establishing and operating such a system, , the potential benefits of central control, deployment layouts limitations, and operational configurations.

Further Explanation of this concept:

1. **Logstash:** Logstash seems to be a powerhouse for consuming information into an automated data optimization procedure. It needs to gather, refine, and integrate log information into an appropriate layout for use in storage and analysis.
2. **Kibana:** Users can communicate with the categorized log data while using visual representations and dashboarding access control Kibana. It appears to offer a graphicuser interface for designing visualizations, operating concerns, and collecting information. Kibana enables users to construct personal dashboards that showcase developments, errors, and structures in log data. Dashboards that integrate significant information should provide immediate knowledge and statements.
3. **Elasticsearch:** Both structured and unstructured information can be organized and tracked down t using Elasticsearch, a reliable and consistent listing and data management generator. It functions as the primary archive for log cloud services in the context of centralized log management. The customized log data is listed by Elasticsearch, which ultimately results in a searchable database offering immediate and adaptable data gathering. The speed and reliability of log data retrieval are strengthened by filtering.
4. **Data Normalization and Transformation:** Logstash directly converts supervised or unsupervised log data into something like a required format that has been adapted for efficient browsing and indexing in Elasticsearch. It might just require extracting essential data, extracting timestamps, and making the necessary arrangements.

With efficient log management, management will monitor behaviors and developments in real time. Users can set up alerts to warn them of situations that need to be fixed quickly and should be addressed to their consideration.

**Windows Event Correlation Process:**

Windows Event Correlation has been the technique for characterizing and finding links and correlations within and between running on windows events to notice advanced and complex attacks or cybercriminals. This strategic approach tends to strengthen the ability to define advanced and powerful attacks that may imply a cycle of associated mishaps rather than solo attacks. Here's a brief representation of the possibility [5].

1. **Gathering Event Data:** The method involves organizing event information from different Windows environment sources. Windows Event Logs monitor the performance of several different occurrences, which would include web applications, application behaviors, and operating systems.
2. **. Event Correlation rules:** Implementing guidelines for event correlation involves understanding correlations between incidences that might indicate malicious or unauthorized behavior. These regulations were created using information about normal network threats and attack procedures.
3. **Event Sources and Types:** Events in several categories, which would include joining materials, wireless routers, user usernames and passwords, and authority changes, offer insights on security events. Many aspects, including the operating system, applications, and security systems, provide events.
4. **Temporal and Logical Correlation:** Correlation among events is primarily focused on both rational and reasonable and empirical (time-related) aspects. Because even though logical similarity appears in the associations among both actions in terms of cause and effect, dynamical connection validates the timing and breakdown of occurrences.
5. **Recognition of patterns:** To find established patterns that correspond to well-known attack scenarios, correlation engines scan event data. These trends may also include predefined cycles of flexion and extension, password cracking, or security breaches.
6. **Event Chains and Sequences:** Optimized cyberattacks often contain a range of situations recognized as malicious code chains. Event effect is provided because of certain event structures that could also refer to a multi-stage assault.
7. **Alert generation:** A signal is managed to be produced if a linkage rule discovers a behavior or collection of occurrences that correspond to the description of a cyberattack context. The security team will observe and adapt more thoroughly as a side effect of this alert.
8. **Incident investigation:** Correlated factors have helped security personnel identify the nature and severity of an event occurring by presenting a better understanding of attack patterns and deadlines.
9. **Contextual Evaluation:** The circumstances of an event that occurs are addressed when relating events. Regarding making properly rational choices concerning the potential of an attack, this involves details regarding the user, the system, and the environment.
10. **Reducing False Positives:** Cutting Down on False Positives By analyzing many changes in practice, effective linkage rules tend to eliminate random errors. This guarantees that alarms are only increased when a clear and present danger is more probable.

The researchers behind the research [5] go through how hack establishments sequence of events together, something that accounts for a lengthy malicious activity spotted that uses reporting and control technologies on Windows event data. Developers would perhaps be processing information algorithms, highlight the consistent and accurate investigation procedures, or processes they were using to optimize cybersecurity in Windows systems.

**Chapter 4: Artefact Design Methodology**

Testing the artifact with users is an essential step to ensure that it meets their needs and expectations. Here's a flow on how to effectively test the artifact with users:

**1. Define Test Goals:**

Clarify the goals of the testing phase. Determine what specific aspects of the artifact you want to test, such as the accuracy of data, usability of dashboards, effectiveness of alerts, and overall user satisfaction.

**2. Identify Test Users:**

Select a diverse group of users who represent different roles and responsibilities within your organization. This can include IT administrators, security analysts, managers, and other relevant stakeholders.

**3. Prepare Test Data:**

Depending on the nature of your artifact, you might need to prepare test data that simulates real-world scenarios. For instance, if you're working with Windows Event Log data in a CSV format, ensure the test data mimics actual log entries.

**4. Conduct User Training:**

Before the testing phase, provide brief training to the test users. Show them how to access the artifact, run queries, interact with visualizations, set up alerts, and navigate through the dashboards.

**5. Test Scenarios:**

Design a set of test scenarios that users can follow to evaluate different aspects of the artifact. These scenarios should cover common use cases and potential edge cases. For example:

- Scenario 1: Check for failed login attempts and visualize trends over the past week.

- Scenario 2: Set up an alert to notify when a specific event code occurs.

- Scenario 3: Investigate a particular security incident by drilling down into relevant events.

**6. Observe and Collect Feedback:**

During the testing phase, ask users to execute the test scenarios. Observe their interactions and gather feedback on their experience. Pay attention to usability issues, performance concerns, and any unexpected behaviors.

**7. Gather Feedback:**

Collect both qualitative and quantitative feedback from users. Use surveys, interviews, or direct discussions to gather insights about their likes, dislikes, and suggestions for improvement.

**8. Document Issues:**

Document any issues or bugs that users encounter during testing. Include detailed information about the problem, the steps to reproduce it, and any error messages.

**9. Address Issues:**

After testing, analyze the feedback and identify issues. Prioritize them based on their impact and severity. Work with your development or implementation team to address these issues promptly.

**10. Iterative Testing:**

Make the necessary fixes and improvements based on the feedback received. Conduct additional rounds of testing with users to ensure that the changes have resolved the reported issues and enhanced the user experience.

**11. User Acceptance Testing (UAT):**

Before finalizing the artifact, consider a formal User Acceptance Testing (UAT) phase where users validate that the implemented changes meet their requirements. This is the last step before deploying the artifact to a wider audience.

**12. Communicate Changes:**

Keep users informed about the changes made based on their feedback. Let them know how their input contributed to the artifact's improvements.

**13. Training and Support:**

Provide updated training to users to showcase the changes and improvements that have been implemented based on their feedback.

**14. Deploy Final Artifact:**

Once testing and improvements are complete, deploy the final version of the artifact for regular use.

**15. Continuous Improvement:**

Continue to gather user feedback even after deployment. Use their insights to iteratively improve the artifact over time.

Remember that user testing is a collaborative process that aims to make the artifact more user-friendly, effective, and aligned with user needs. It's crucial to maintain open communication with users and create an environment where their input is valued and integrated into the development process.

Capturing and interpreting results from testing is crucial for understanding how well your artifact is performing and identifying areas for improvement. Here's how to effectively capture and interpret testing results:

**1. Set Clear Metrics:**

Define specific metrics and key performance indicators (KPIs) that you'll use to evaluate the success of your artifact. These could include accuracy of data, response times, user satisfaction scores, and more.

**2. Document Test Scenarios:**

Document each test scenario you designed for the testing phase. Include the expected outcomes, steps to reproduce, and any specific criteria for success.

**3. Capture Observations:**

As users execute test scenarios, observe their interactions with the artifact. Document their actions, choices, and any difficulties they encounter. Note down any positive feedback as well.

**4. Collect Feedback:**

Gather qualitative feedback from users through surveys, interviews, or focus groups. Ask about their overall experience, usability, performance, and any issues they encountered.

**5. Document Issues:**

Keep a detailed record of any issues, bugs, or glitches that users encounter during testing. Include information about the nature of the issue, the context in which it occurred, and any error messages.

**6. Analyze Metrics:**

Analyze the quantitative metrics you set beforehand. Compare the actual results with your predefined success criteria. For example, if you aimed for a certain level of accuracy in data extraction, assess whether that goal was met.

**7. Assess Usability:**

Assess the usability of the artifact by observing how easily users navigate through the interface, run queries, interact with visualizations, and set up alerts. Look for patterns of usability challenges or confusion.

**8. Identify Patterns:**

Look for patterns in the feedback and issues reported by different users. Are there common pain points or recurring themes? Identifying patterns helps you prioritize improvements.

**9. Triangulate Feedback:**

Cross-reference qualitative feedback with quantitative metrics. For instance, if users report slow response times, check if this aligns with performance metrics you've measured.

**10. Categorize Feedback:**

Categorize feedback into different areas such as usability, performance, accuracy, and overall satisfaction. This helps you focus on specific aspects during the improvement phase.

**11. Prioritize Improvements:**

Based on the severity and impact of issues and user feedback, prioritize which areas need improvement first. Focus on addressing critical issues that significantly affect usability or functionality.

**12. Root Cause Analysis:**

For issues that arise, conduct a root cause analysis to understand why they occurred. Was it a technical glitch, a usability flaw, or a misconfiguration? This analysis informs how to fix the issues effectively.

**13. Plan Enhancements:**

Develop an action plan for addressing the identified issues and improving the artifact. Outline the steps needed to fix bugs, enhance usability, and optimize performance.

**14. Iterative Improvement:**

Use the captured results and feedback as a foundation for iterative improvement. Implement the necessary changes, and conduct further testing to validate that the improvements have been successful.

**15. Communicate Results:**

Communicate the testing results, including the captured observations, feedback, and improvements, to stakeholders, users, and the development team. Transparency is key for aligning everyone's expectations.

**16. Documentation:**

Document the testing process, results, and the actions taken for improvements. This documentation is valuable for reference in the future and can provide insights for similar projects.

Interpreting testing results involves analyzing both quantitative and qualitative data to gain a holistic understanding of your artifact's performance and user experience. It guides you in making informed decisions about enhancements and adjustments that will ultimately lead to a more effective and user-friendly artifact.

# **Chapter 5: Implementation**

1. Environment Setup:

Ensure you have the required hardware and software resources ready. Since you're working with a CSV dataset, you won't need Windows hosts or Universal Forwarder. You'll need a server or virtual machine to host your Splunk instance. Check Splunk's hardware requirements based on your data volume and expected usage.

Hardware Resources:

|  |  |
| --- | --- |
| Property | Description |
| Manufacturer | HP |
| Model | Pavilion |
| Architecture | x64 based |
| Operating System | Windows 11 |
| Processor | Intel(R) Core (TM) i7-8550U CPU @ 1.80GHz 1.99 GH GHz |
| RAM | 8 GB |
| Storage | 1 TB |

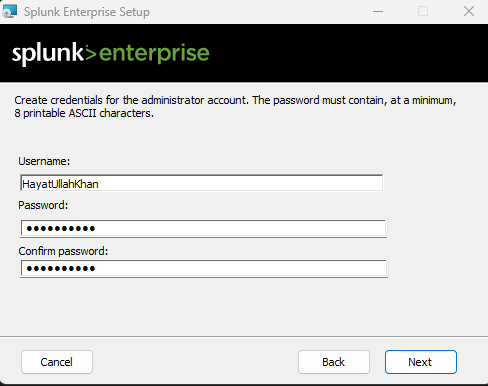
**2. Splunk Installation:**

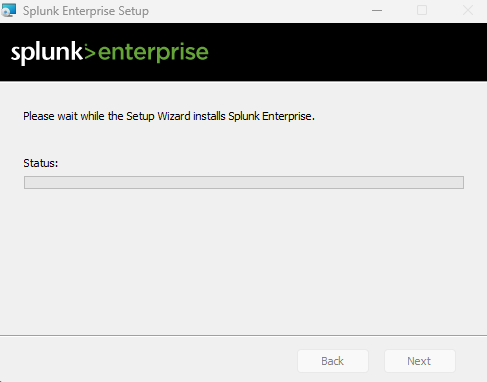
Install Splunk Enterprise on the chosen server or virtual machine. Follow these steps:

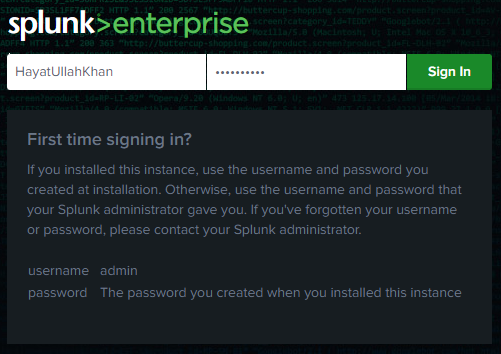
- Download the Splunk Enterprise installation package for your operating system from the Splunk website. set up an account on Splunk Cloud

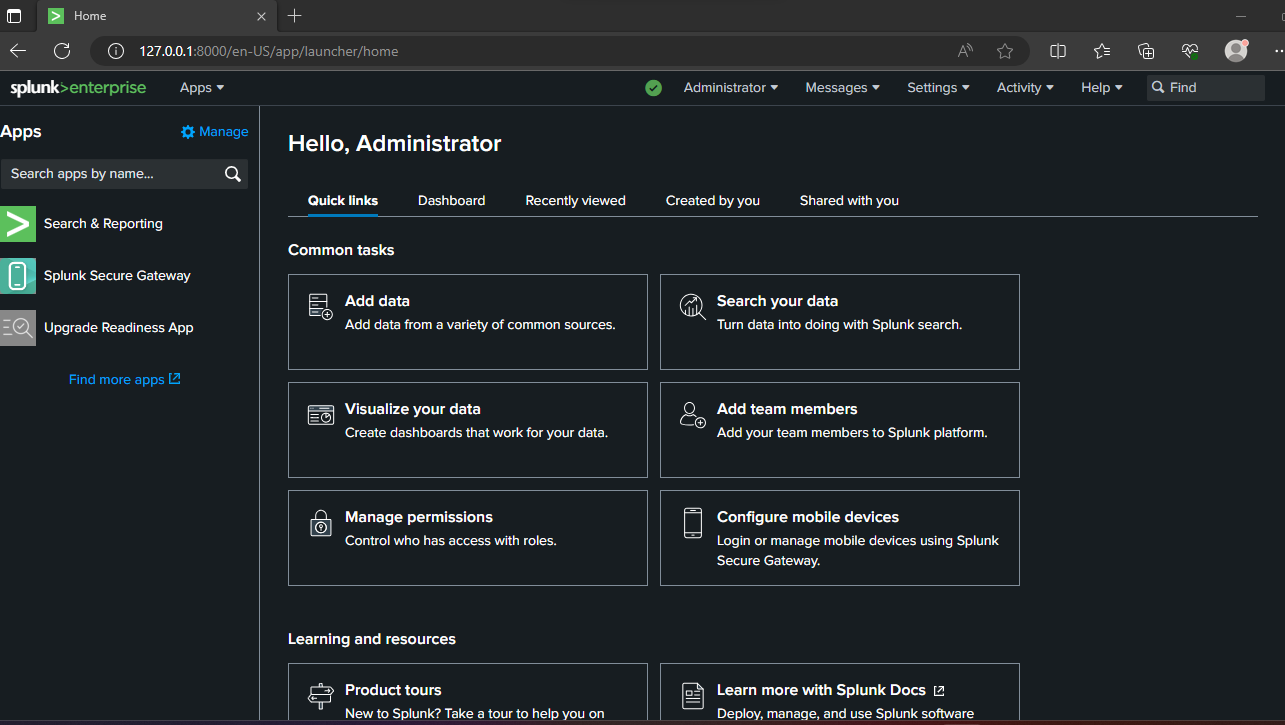
- Run the installer and follow the prompts. Choose a directory for installation.

- Start Splunk after installation completes. Access the Splunk Web interface by opening a web browser and navigating to `http://localhost:8000`.









**Add data:**

Certainly, here's a detailed explanation of each step for implementing the artifact using Windows Event Log data stored in a CSV dataset:

**3. Configure Data Inputs:**

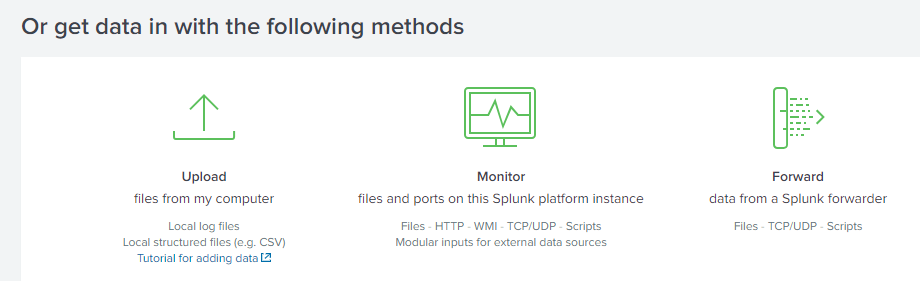
Since you have a CSV dataset, you'll need to upload it into Splunk. Follow these steps:

- Log in to the Splunk Web interface using the default credentials (admin/admin).

- Navigate to "Settings" > "Data inputs."

- Choose "Files & Directories" and then "New."

- Browse for your CSV dataset and specify the source type (e.g., `csv\_windows\_event`).



**4. Create Indexes:**

You can create an index to store the CSV data:

- Navigate to "Settings" > "Indexes."

- Click "New Index" and provide a name (e.g., `windows\_event\_logs`).

- Configure retention settings and other options as needed.

A screenshot of a computer

Description automatically generated

**4. Configure Source Types:**

Configure a source type for your CSV data:

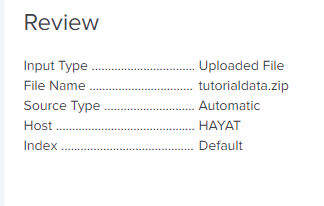
- Navigate to "Settings" > "Source types."

- Click "Add new" and provide a name (e.g., `csv\_windows\_event`).

- Define field extractions, timestamp format, and other settings.

A screenshot of a computer

Description automatically generated



**5. Writing Search Queries:**

Design search queries to extract relevant information from the Windows Event Log data. Splunk's Search Processing Language (SPL) allows you to filter, transform, and aggregate data. Build search queries to extract information from your CSV data:

- Use the Splunk Search & Reporting app to construct queries.

- For example, to retrieve all events:

`index=windows\_event\_logs sourcetype=csv\_windows\_event`

**Consider the following example queries:**

1. View All Events:\*\* Retrieve all Windows Event Log events for analysis.

*index=<your\_index> sourcetype="wineventlog:security"*

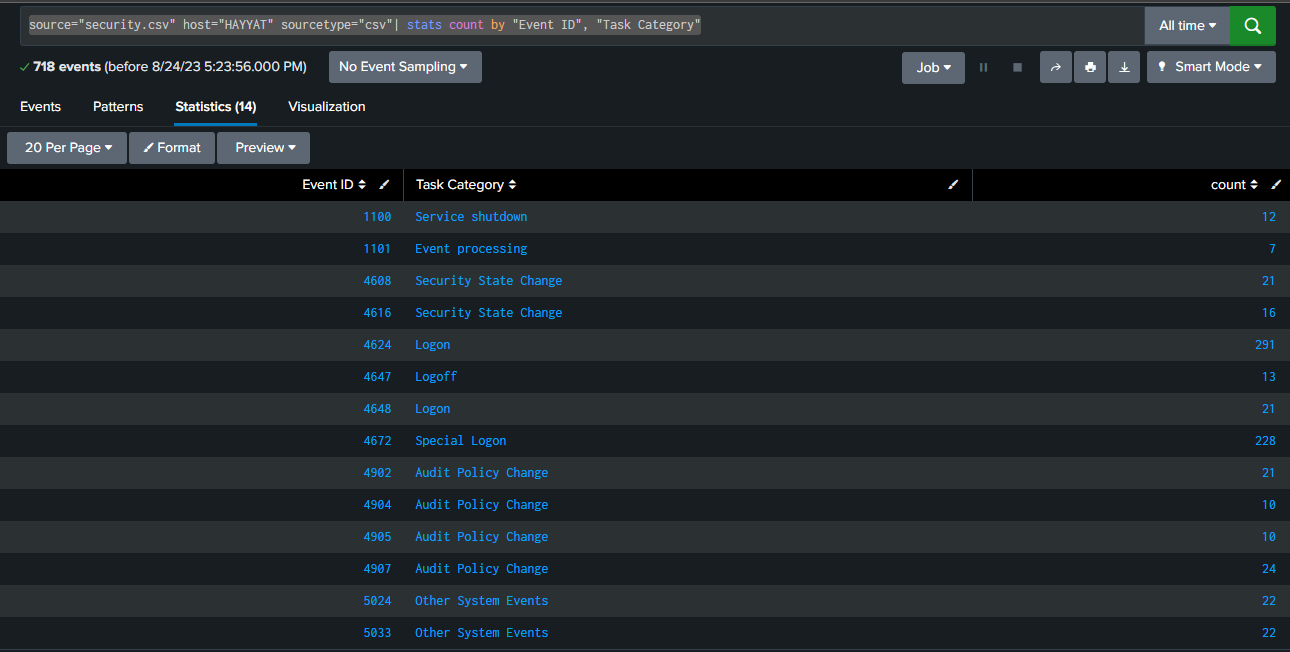
*| table \_time, EventCode, EventID, Message*

Filter by Event Type:\*\* Retrieve specific event types, e.g., successful logins.

*index=<your\_index> sourcetype="wineventlog:security" EventCode=4624*

*| table \_time, Account\_Name, Source\_Network\_Address*

source="security.csv" host="HAYYAT" sourcetype="csv"| stats count by "Event ID", "Task Category"



**6. Build Visualizations:**

Create visualizations to represent the extracted data in a meaningful way. Splunk offers various visualization options such as charts, tables, and dashboards. Create visualizations based on the search queries:

- Navigate to "Search & Reporting" > "Visualizations."

- Choose a visualization type (e.g., "Line Chart," "Table," "Pie Chart") and configure settings based on your query results.

**Time Chart Visualization:**

Create a time chart to visualize the trend of successful login events over time.

index=<your\_index> sourcetype="wineventlog:security" EventCode=4624

| timechart span=1h count by host

source="security.csv" host="HAYYAT" sourcetype="csv" "Event ID"=4624 | timechart count

A graph on a computer screen

Description automatically generated

source="security.csv" host="HAYYAT" sourcetype="csv" | timechart count by "Event ID"

A screen shot of a computer

Description automatically generated

**7. Design Dashboards:**

Combine visualizations into dashboards for easy monitoring and interaction. Dashboards provide a consolidated view of the data and insights.

Assemble visualizations into dashboards:

- Navigate to "Create" > "Dashboard" in the Splunk Web interface.

- Add panels by selecting visualizations or search queries.

- Organize and format the dashboard layout to convey insights effectively.

**8. Set Up Alerts:**

Configure alerts to receive notifications when specific events occur. For example, you can set up an alert to notify you when a certain number of failed login attempts is detected.

Set up alerts based on search criteria:

- Navigate to "Settings" > "Alerts."

- Click "Create Alert" and specify the trigger conditions, frequency, and notification options (e.g., email alerts).

**9. Correlation Searches:**

Create correlation searches to identify patterns or anomalies across different data sources. This can help detect security incidents or operational issues.

**10. Scheduled Reports:**

Automate the generation and distribution of reports based on the search results. This is useful for sharing insights with stakeholders.

**11. Advanced Queries:**

Consider advanced queries that involve statistical analysis, anomaly detection, and machine learning capabilities that Splunk provides.

**12. Maintain and Optimize:**

Regularly review and optimize your queries, indexes, and dashboards to ensure they remain relevant and performant as your data and requirements evolve.

**13. Testing:**

Thoroughly test your architecture and deployment. Ensure that your search queries, visualizations, and alerts are working as expected. Simulate different scenarios to ensure the system can handle peak loads.

**14. Documentation:**

Document your deployment architecture, configuration settings, search queries, and any customizations you've made. This documentation will be valuable for troubleshooting, maintenance, and knowledge sharing.

**15. Monitoring and Maintenance:**

Implement monitoring tools to keep an eye on the health and performance of your Splunk instance. Regularly review and optimize your search queries, indexes, and visualizations. Perform routine maintenance tasks such as software updates and security patches.

Here's a single scenario in narrative form where a SOC analyst uses the provided queries to investigate potential security incidents:

**Chapter 6: Results and Evaluation**

As a Security Operations Center (SOC) analyst, Emily's day starts with a fresh cup of coffee and a quick review of the overnight activities. Today, her primary focus is on investigating user authentication activities, account management, and system health within the Windows environment.

Emily's attention is immediately drawn to a spike in successful logins (Event ID 4624). She crafts a Splunk query to retrieve the details of these logins, analyzing logon types and authentication packages to ensure legitimate access. A few entries catch her eye due to their unusual logon types at odd hours. She marks these for further investigation.

Later in the morning, Emily shifts her focus to account management and policy changes. She identifies that a new user has been added to a sensitive security group (Event ID 4907). She quickly pulls up relevant information using Splunk, noting the target account and associated group. With this information, she collaborates with the access control team to verify the change and assess its authorization.

As the day progresses, Emily notices a change in user rights assignments (Event ID 4904). Some policies have been altered. She investigates the changes, cross-referencing with change management records. A closer look reveals a potential anomaly where a non-admin user gained temporary elevated rights. Emily escalates this to her team lead for immediate action, suspecting a possible privilege escalation attempt.

After lunch, Emily switches her attention to system health. She spots an unusual pattern of frequent service starts (Event ID 7024). Using Splunk, she queries for these events, focusing on the services' statuses and associated users. While most are legitimate, she uncovers a few cases where non-standard services were started. She coordinates with the system administrators to ensure these services are authorized.

Before wrapping up for the day, Emily reviews system shutdown and startup events (Event IDs 1100 and 1101). She identifies a shutdown event that occurred without proper notification and investigates further. After discussions with the IT team, she confirms that the shutdown was part of a planned maintenance. She documents the event for future reference and incident reports.

As the day comes to an end, Emily reflects on her investigations and actions. Her diligent analysis using Splunk's powerful queries has helped the SOC maintain a secure environment. She's confident that her efforts have contributed to detecting potential threats, preventing unauthorized access, and ensuring the overall health of the system.

In this narrative scenario, the SOC analyst, Emily, employs the provided Splunk queries to investigate a range of security-related events, from user authentication to account management and system monitoring. Her actions showcase her dedication to maintaining a secure and well-functioning IT environment.

Absolutely, here are complete scenarios for each of the three use cases, tailored for a Security Operations Center (SOC) analyst using Splunk.

Use Case 1: User Authentication Analysis

Scenario: Uncovering Suspicious User Authentication Activities

Background: The SOC analyst is responsible for monitoring and investigating user authentication activities within the Windows environment to identify potential security threats and unauthorized access.

Steps:

**1. Initial Search for Successful Logins:**

- Query:

```splunk

index=your\_index sourcetype=WinEventLog:Security EventCode=4624

| table \_time, host, user, Logon\_Type, Logon\_Process, Authentication\_Package

- Action: Run the query to retrieve successful logins, focusing on fields such as user, logon type, and authentication package.

**2. Analyze Successful Logins:**

- Investigate the retrieved successful logins to verify if they are expected and legitimate. Pay attention to any unusual logon types or authentication packages.

**3. Identify Privilege Escalation Attempts:**

- Query:

index=your\_index sourcetype=WinEventLog:Security EventCode=4648

| table \_time, host, user, New\_Process\_Name, Target\_User, Target\_Logon\_Id

- Action: Execute the query to identify potential privilege escalation attempts. Look for suspicious processes, target users, and logon IDs.

**4. Detect Failed Logon Attempts:**

- Query:

index=your\_index sourcetype=WinEventLog:Security EventCode=4647

| table \_time, host, user, Failure\_Reason

- Action: Run the query to uncover failed logon attempts. Investigate the failure reasons and determine if they indicate potential malicious activity.

**5. Investigation and Response:**

- For any suspicious or unauthorized activities, cross-reference with other sources and conduct further investigation using additional security tools.

- Depending on the severity, either escalate to higher-level analysts or take immediate response actions following the organization's incident response plan.

Use Case 2: Account Management and Policy Changes

Scenario: Detecting Unauthorized Account and Policy Changes

Background: The SOC analyst needs to monitor account management activities and policy changes to ensure security policies are upheld and unauthorized changes are identified.

Steps:

**1. Monitoring Account Additions to Groups:**

- Query:

index=your\_index sourcetype=WinEventLog:Security EventCode=4907

| table \_time, host, Target\_Account, Target\_Type, Group\_Name

- Action: Execute the query to monitor additions to local security groups. Focus on target accounts and associated groups

**2. Identify User or Group Creation:**

- Query:

index=your\_index sourcetype=WinEventLog:Security EventCode=4902

| table \_time, host, Subject\_User, Object\_Name, Object\_Type

- Action: Run the query to identify user or group creations. Pay attention to the subject user, object name, and type.

**3. Track Changes to User Rights Assignment:**

- Query:

index=your\_index sourcetype=WinEventLog:Security EventCode=4904

| table \_time, host, user, Policy\_Key, Changed\_Value

- Action: Execute the query to track changes to user rights assignment. Look for altered policy keys and changed values.

**4. Monitor Changes to Security Policies:**

- Query:

index=your\_index sourcetype=WinEventLog:Security EventCode=4905

| table \_time, host, user, Policy\_Target, Policy\_ID, Policy\_Changes

- Action: Run the query to monitor changes to security policies. Investigate the policy targets, IDs, and changes made.

**5. Investigation and Response:**

- Compare detected changes with approved change requests or policies to determine whether the changes are authorized.

- If unauthorized changes are identified, follow the organization's incident response procedures to contain, eradicate, and recover from the incident.

Use Case 3: Service and System Monitoring

Scenario: Ensuring Service Integrity and System Health

Background: The SOC analyst is responsible for monitoring system services and events to maintain system integrity and respond to potential issues promptly.

Steps:

**1. Identify Windows Service Starts:**

- Query:

index=your\_index sourcetype=WinEventLog:Security EventCode=7024

| table \_time, host, user, Service\_Name, Service\_Status

```

- Action: Execute the query to identify Windows service starts. Pay attention to the service name and its status.

**2. Detect Changes in Service Configuration:**

- Query:

index=your\_index sourcetype=WinEventLog:Security EventCode=5033

| table \_time, host, user, Service\_Name, Config\_Changes

- Action: Run the query to detect changes in service configuration. Investigate the service name and configuration changes made.

**3. Monitor System Shutdown Events:**

- Query:

index=your\_index sourcetype=WinEventLog:Security EventCode=1100

| table \_time, host, user, Shutdown\_Type, Comment

- Action: Execute the query to monitor system shutdown events. Look for shutdown types and any associated comments.

**4. Investigate System Startup Events:**

- Query:

index=your\_index sourcetype=WinEventLog:Security EventCode=1101

| table \_time, host, user, Boot\_Time, Boot\_Type

- Action: Run the query to investigate system startup events. Analyze boot times and boot types.

**5. Investigation and Response:**

- For any unusual service starts, changes, or system events, consult system documentation and relevant teams to verify if they are legitimate.

- If suspicious activities are identified, assess the impact and escalate as necessary according to incident response protocols.

In each scenario, the SOC analyst performs a series of steps to monitor, detect, analyze, and respond to potential security incidents and anomalies related to the specified Windows event IDs. Keep in mind that the scenarios are adaptable to your organization's specific processes, data sources, and security policies.

[1] Schauland, D., Jacobs, D., Schauland, D. and Jacobs, D., 2016. Managing the windows event log. Troubleshooting Windows Server with PowerShell, pp.17-33.

[2] J. Dwyer and T. M. Truta, "Finding anomalies in windows event logs using standard deviation," 9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing, Austin, TX, USA, 2013, pp. 563-570, doi: 10.4108/icst.collaboratecom.2013.254136.

[3] Rozinat, A., De Jong, I.S.M., Günther, C.W. and van der Aalst, W.M., 2007. Process mining of test processes: A case study.

[4] Ahmed, F., Jahangir, U., Rahim, H. and Ali, K., 2020, February. Centralized Log Management Using Elasticsearch, Logstash and Kibana. In 2020 International Conference on Information Science and Communication Technology (ICISCT) (pp. 1-7). IEEE.

[5] Yamiun, M.M., Katt, B. and Gkioulos, V., 2020. Detecting windows-based exploit chains by means of event correlation and process monitoring. In Advances in Information and Communication: Proceedings of the 2019 Future of Information and Communication Conference (FICC), Volume 2 (pp. 1079-1094). Springer International Publishing.

# **Reference**

[1] Sahoo, P.K., Chottray, R.K. and Pattnaiak, S., 2012. Research issues on windows event log. *International Journal of Computer Applications*, *41*(19).

[2] Ibrahim, N.M., Al-Nemrat, A., Jahankhani, H. and Bashroush, R., 2011, August. Sufficiency of windows event log as evidence in digital forensics. In *International Conference on e-Democracy* (pp. 253-262). Berlin, Heidelberg: Springer Berlin Heidelberg.

[3] Codona, B., 2007. Analysis and Evaluation of the Windows Event Log for Forensic Purposes. In *Honour Project, Edinburgh Napier Univ.*

[4] Do, Q., Martini, B., Looi, J., Wang, Y. and Choo, K.K., 2014. Windows event forensic process. In *Advances in Digital Forensics X: 10th IFIP WG 11.9 International Conference, Vienna, Austria, January 8-10, 2014, Revised Selected Papers 10* (pp. 87-100). Springer Berlin Heidelberg.

[5] He, S., Zhu, J., He, P. and Lyu, M.R., 2020. Loghub: A large collection of system log datasets towards automated log analytics. *arXiv preprint arXiv:2008.06448*.

[6] SearchWindowsServer. (n.d.). What is Windows event log? - Definition from WhatIs.com. [online] Available at: https://www.techtarget.com/searchwindowsserver/definition/Windows-event-log.

[7] What Is a Windows Event Log? - IT Glossary | SolarWinds. (n.d.). Www.solarwinds.com. Retrieved July 28, 2023, from <https://www.solarwinds.com/resources/it-glossary/windows-event-log>.

[8] www.manageengine.com. (n.d.). *Log Visualization and Analysis Tool - Read and Analyze your Logs Online | EventLog Analyzer*. [online] Available at: https://www.manageengine.com/products/eventlog/log-visualization-tool.html [Accessed 28 Jul. 2023].

[9] crowdstrike.com. (n.d.). *What is an Application Log? Definition and Types - CrowdStrike*. [online] Available at: <https://www.crowdstrike.com/cybersecurity-101/observability/application-log/>.

[10] www.solarwinds.com. (n.d.). *What Is a Windows Event Log? - IT Glossary | SolarWinds*. [online] Available at: https://www.solarwinds.com/resources/it-glossary/windows-event-log.

‌

‌

‌