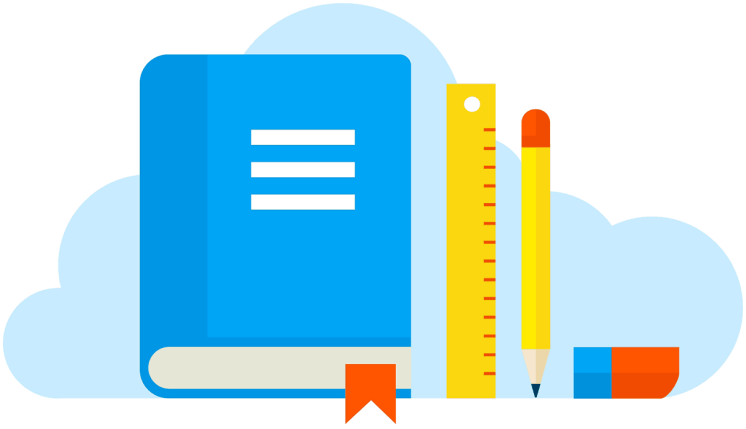

SOC Level 2

Date: 08/09/2025

Intro to Logs

- Expanding Perspectives: Logs as Evidence of Historical Activity
    - When log data is aggregated, analysed, and cross-referenced with other sources of information, it becomes a potent investigation tool
        i. **What** happened?
        ii. **When** did it happen?
        iii. **Where** did it happen?
        iv. **Who** is responsible?
        v. **Were** their actions **successful**?
        vi. **What** was the result of their action?
- Types, Formats, and Standards
    - Log Collection is an essential component of log analysis, which involves the aggregation of logs from a diverse number of sources.
    - Crucial to maintain the systems time accuracy during logging.
    - Utilising the Network Time Protocol (NTP) is a method to achieve this synchronization and ensure the integrity of the timeline stored in the logs.
    - Step-by-step process
        i. Identify Sources: List all potential log sources, such as servers, databases, applications, and network devices

ii. Choose a log collector: Opt for a suitable log collector tool or software that aligns with your infrastructure
iii. Configure Collection Parameters: Ensure that time synchronization is enabled through NTP to maintain accurate timelines, adjust settings to determine which events to log at what intervals, and prioritize based on importance
iv. Test Collection: Once configured, run a test to ensure logs are appropriately collected from all sources
v. Performing NTP-based time synchronization in practice using pool.ntp.org to find an NTP server is best.
vi. Time synchronization can be performed automatically on linux-based systems or manually initiated by executing ntpdate pool.ntp.org

- Log Management ensures that logs are stored securely, organized systematically and are ready for swift retrieval.
- Step-by-step for log management:
  i. Storage: Decide on a secure storage solution, considering factors like retention period and accessibility
  ii. Organization: Classify logs based on their source, type, or other criteria for easier access later
  iii. Backup: Regularly back up your logs to prevent data loss
  iv. Review: Periodically review logs to ensure they are correctly stored and categorized
- Log Centralization is important for log access, in-depth analysis, and rapid incident response
- A unified system allows for efficient log management with tools that offer real-time detection, automatic notifications, and seamless integration with incident management systems
- Step-by-step process for centralizing logs:
  i. Choose a centralised system: Opt for a system that consolidates logs from all sources, such as the Elastic Stack or Splunk.
  ii. Integrate sources: Connects all your lo sources to this centralized system
  iii. Set Up Monitoring: Utilize tools that provide real-time monitoring and alerts for specified events

iv. Integration with Incident Management: Ensure that your centralized system can integrate seamlessly with any incident management tools or protocols

- Practical Activity: Log collection with rsyslog
  i. This activity aims to introduce rsyslog and demonstrate how it can enhance the centralisation and management of logs. As part of the collection process, we will configure rsyslog to log all sshd messages to a specific file, such as /var/log/websrv-02/rsyslog_sshd.log. The steps below can be followed to achieve this:
     1. **Open a Terminal.**
     2. **Ensure rsyslog is Installed:** You can check if rsyslog is installed by running the command: `sudo systemctl status rsyslog`
     3. **Create a Configuration File:** Use a text editor to create the following configuration file: `gedit /etc/rsyslog.d/98-websrv-02-sshd.conf`, `nano /etc/rsyslog.d/98-websrv-02-sshd.conf`, `vi /etc/rsyslog.d/98-websrv-02-sshd.conf`, or `vim /etc/rsyslog.d/98-websrv-02-sshd.conf`
     4. **Add the Configuration:** Add the following lines in `/etc/rsyslog.d/98-websrv-02-sshd.conf` to direct the sshd messages to the specific log file:
        $FileCreateMode 0644
     5. :programname, isequal, "sshd"
        /var/log/websrv-02/rsyslog_sshd.log
     6. **Save and Close the Configuration File.**
     7. **Restart rsyslog:** Apply the changes by restarting rsyslog with the command: `sudo systemctl restart rsyslog`
     8. **Verify the Configuration:** You can verify the configuration works by initiating an SSH connection to localhost via `ssh localhost` or by checking the log file after a minute or two.

- Storage Retention, and Deletion
  - Log Storage

i. Storage location is dependent upon a few factors:
    1. **Security Requirements:** Ensuring that logs are stored in compliance with organisational or regulatory security protocols.
    2. **Accessibility Needs:** How quickly and by whom the logs need to be accessed can influence the choice of storage.
    3. **Storage Capacity:** The volume of logs generated may require significant storage space, influencing the choice of storage solution.
    4. **Cost Considerations:** The budget for log storage may dictate the choice between cloud-based or local solutions.
    5. **Compliance Regulations:** Specific industry regulations governing log storage can affect the choice of storage.
    6. **Retention Policies:** The required retention time and ease of retrieval can affect the decision-making process.
    7. **Disaster Recovery Plans:** Ensuring the availability of logs even in system failure may require specific storage solutions.

- Log Retention
    i. A reasonable balance between retaining logs for potential future needs and the storage cost comes to understanding the different types of storage concepts:
        1. **Hot Storage:** Logs from the past **3-6 months** that are most accessible. Query speed should be near real-time, depending on the complexity of the query.
        2. **Warm Storage:** Logs from **six months to 2 years**, acting as a data lake, easily accessible but not as immediate as Hot storage.
        3. **Cold Storage:** Archived or compressed logs from **2-5 years**. These logs are not easily accessible and are usually used for retroactive analysis or scoping purposes.
- Log Deletion
    i. Must be performed carefully to avoid removing logs that could still be of value
    ii. Backup log files before deletion

iii. Have a well-defined deletion policy to ensure compliance with data protection laws and regulations, log deletion helps to:
    1. Maintain a manageable size of logs for analysis.
    2. Comply with privacy regulations, such as GDPR, which require unnecessary data to be deleted.
    3. Keep storage costs in balance.
- Best Practices: Log Storage, Retention and Deletion
    i. Determine the storage, retention, and deletion policy based on both business needs and legal requirements
    ii. Regularly review and updated the guidelines per changing conditions and regulations
    iii. Automate the storage, retention, and deletion process to ensure consistency and avoid human errors
    iv. Encrypt sensitive logs to protect data
    v. Regular backups should be made, especially before deletion
- Practical Activity: Log Management with logrotate
    i. Logrotate automates log file rotation, compression, and management
    ii. Allows automatic rotation, compression and removal of log files

- Hands-On Exercise: Log Analysis process, tools, and techniques
    - Log Analysis Process
        i. Data Sources
            1. Systems or applications configures to log system events or user activities, origin of logs
        ii. Parsing
            1. Breaking down the log data into more manageable and understandable components, logs come in various formats depending on source, essential to parse the logs to extract valuable info
        iii. Normalisation
            1. Standardizing parsed data, bringing the various log data into a standard format
        iv. Sorting

1. Allows for efficient data retrieval and identification of patterns, can be sorted by time, source, event type, severity, and any other parameter present in the data.
2. Critical in identifying trends and anomalies that signal operational issues or security incidents

v. Classification
   1. Assigning categories to the logs based on their characteristics
   2. Classifying log files allows for quick filter and focus on the logs that matter for analysis.
   3. Automated classifications using machine learning can significantly enhance this process, helping to identify potential issues or threats that could be overlooked

vi. Enrichment
   1. Log enrichment adds context to logs to make them more meaningful and easier to analyse
   2. Adding geographical data, user details, threat intel, or data from other sources that can provide a complete picture of the event
   3. Enrichment makes logs more valuable, enabling analysts to make better decisions and more accurately respond to incidents

vii. Correlation
   1. Linking related records and identifying connections between log entries
   2. Helps detect patterns and trends, making understanding complex relationships between various log events easier
   3. Correlation is critical in determining security threats or system performance issues that might remain unnoticed

viii. Visualization
   1. Represents log data in a graphical format like charts, graphs, or heat maps.
   2. Visually presenting data makes recognizing patterns, trends, and anomalies easier

3. Provides intuitive way to interpret large volumes of log data, making complex info more accessible and understandable

ix. Reporting

1. Summarizes log data into structured formats to provide insights, support decision-making, or meet compliance requirements.

- Log Analysis Tools
  - i. Linux-based systems can employ default tools like cat, grep, sed, sort, uniq, and awk, along with sha256sum for hashing log files
  - ii. Windows-based systems can utilize EZ-Tools and the default cmdlet Get_FileHash for similar purposes
  - iii. Proper acquisition should be observed by taking the log files hash during collection to ensure its admissibility in court.
- Log Analysis Techniques
  - i. Methods or practices used to interpret and derive insights from log data:
    1. **Pattern Recognition:** identifying recurring sequences or trends in log data, detect regular system behaviour or identify unusual activities that may indicate a security threat
    2. **Anomaly Detection:** identifying data points that deviate from the expected pattern
    3. **Correlation Analysis:** helps understand the relationship between various events. It can reveal causation and dependencies between system components and is vital in root cause analysis.
    4. **Timeline Analysis:** Analysing logs over time helps understand trends, seasonalities, and periodic behaviours. It can be essential for performance monitoring and forecasting system loads.
    5. **Machine Learning and AI:** Leveraging machine learning models can automate and enhance various log analysis techniques, such as classification and enrichment. AI can provide predictive insights and help in automating responses to specific events.

6. **Visualisation:** Representing log data through graphs and charts allows for intuitive understanding and quick insights. Visualisation can make complex data more accessible and assist in identifying key patterns and relationships.
7. **Statistical Analysis:** Using statistical methods to analyse log data can provide quantitative insights and help make data-driven decisions. Regression analysis and hypothesis testing can infer relationships and validate assumptions.

- Log Operations
    - Log Configuration Options
        i. a multifaceted operation that addresses security, operational stability, regulatory compliance, and debugging needs.
    - Security Purposes
        i. Logging and configuration for security purposes are typically planned to detect and respond to anomalies and security issues.
            1. Anomaly and threat detection
            2. Logging user authentication data
            3. Ensuring the system's integrity and data confidentiality
    - Operational Purposes
        i. Logging and configuring for operational purposes is usually planned to detect and respond to system errors and identify action points to enhance the system's performance, continuity, and reliability.
            1. Proactively creating reports and notifications for on-system and component status
            2. Troubleshooting
            3. Capacity planning
            4. Service billing
    - Legal Purposes
        i. Logging and configuring for legal purposes is similar to security purposes; it is usually planned to stay compliant and increase the alignment with regulations and obligations.
            1. Alignment with standards, compliance, regulations, and laws
            2. E.g. ISO 27001, COBIT, GDPR, PCI DSS, HIPAA, FISMA
    - Debug Purposes

i. Logging and configuring for debug purposes is usually planned to boost the system's reliability and enhance provided features by discovering the bugs and potential fault conditions.
    1. Increasing visibility for the application debugging
    2. Enhancing efficiency
    3. Speeding up the development process

- Where to Start and What to Do After Deciding the Log Purpose?
    - Questions To Ask In PLanning Meeting/Session
        i. What will you log, and for what (asset scope and logging purpose)?
            1. Is additional commitment or effort required to achieve the purpose (requirements to the purpose)?
        ii. How much are you going to log(detail scope)?
        iii. How much do you need to log?
        iv. How are you going to log(collection)?
        v. How are you going to store collected logs?
            1. Is there a standard, process, legislation, or law that you must comply with due to the data you log?
        vi. How are you going to protect the logs?
        vii. How are you going to analyze collected logs?
        viii. Do you have enough resources and workforce to do logging?
        ix. Do you have enough budget to plan, implement and maintain logging?
- Configuration Dilemma: Requirements Aspirations, Resources, and Investment
    - Each log configuration plan results from a unique analysis of the scope, assets, objectives, requirements, and expectations to be applied.
    - the main source of the dilemma is finding the balance between requirements, scope, details, and price (financial and labour costs, risks, and investment)
    - Meeting specific operational and security requirements (non-negotiable) while also considering the feasibility of improving the capability by implementing additional data and insights.
- Translating “Requirements” and “Aspirations” to operational level
    - two question sets represent two distinct dimensions of logging and analysis:

i. The base part heavily relies on an incident detection mindset. Still, it provides a solid framework for logging and analysis but is reactive. The requirements are a good place to start, but it is primarily helpful against known threats.

ii. The aspirations part is more focused on a threat-hunting mindset. Therefore, it is proactive and requires more resources due to the need to go above and beyond. Therefore, this part is more helpful against advanced and sophisticated threats.

- Logging Principles
  -

Collection	• Define the logging purpose. • Collect what you will need and use. • Do not collect irrelevant data. • Avoid log noise.
Format	• Log at the correct level and detail. • Implement a consistent log format. • Ensure that timestamps in logs are accurate and synchronised.
Archiving and Accessibility	• Define log retention policies and implement them. • Store log data and make sure the important part is available for analysis. • Create backups of stored log data and used systems.
Monitoring and Alerting	• Create alerts and notifications for important and noteworthy cases. • Focus on actionable alerts and avoid noise.

Security	• Protect logs by implementing access controls. • Implement encryption if required. • Use a dedicated log management solution.
Continuous Change	• Logging sources, types, and messages are constantly changing and being updated. ○ Be open to continuous change. • Train your personnel.

- Challenges
  -

Data Volume and Noise	• Having multiple data sources to deal with.
	• Differences in the log volumes created by applications.
	  ○ Some applications generate an insufficient amount of logs.
	  ○ Large-scale applicants could generate massive log volumes.
	• Some logs can provide non-essential data and make the identifying process difficult.

System Performance and Collection	• Log collection can slow down the system's performance. • Systems are not always "state of the art". ○ Some "sensitive" or

	"ancient" systems are impossible to touch. ● Deployment and optimisation challenges. ○ Managing system and agent version updates and synchronisation in large-scale networks is overwhelming.
Process and Archive	● Having multiple data formats to deal with it. ○ Parsing different data sources and formats is time-consuming and error-prone. ● Balancing the log retention can be challenging. ○ Especially when dealing with many compliance regulations and standards.
Security	● Ensuring data security is a task/challenge in itself.
Analysis	● Combining, correlating, and analysing data from multiple sources to understand the context of an incident is a time-consuming process that requires significant computing resources and expertise. ○ Achieving this in real-time is also another challenge in the same scope. ○ Avoiding false

	positives/negatives is overwhelming.
Misc	• Lack of planning and roadmap. • Lack of financial resources/budget. • Lack of implementation scenarios, playbooks, and exercises. • Lack of technical skills to implement, maintain, and analyse. • Focusing on log collection instead of the analysis phase. • Ignoring human factors and potential system errors.

- Where to Go From Here?
  - the main point is adhering to logging principles and proactively addressing challenges.
- Common Mistakes and Best Practice
  - the infamous "if it works, don't touch it!" approach is unacceptable
    i. Learn from mistakes and failures.
    ii. Track the sectoral threat dynamics for the operated sector and conduct regular scope and resilience testing.
    iii. Follow the best practices of industry leaders and experts.
  -

Mistakes **"don'ts"**	Best Practices "dos"
• Logging sensitive information! • Creating logs by yourself.	• Create a suitable log configuration and plan according to your systems.

• Having uncollected logs. • Collecting everything but not analysing. • Collecting logs without proper planning and configuration. • Having systems that lack the planned/required log configuration. • Skipping the scale, testing, and functionality analysis. • Focusing on edges and skipping the internal systems in analysis. • "Searching for what you want to find" and "Not investigating what you see". • Forgetting that the process takes the form of proper planning, management, and analysis.	• Implement testing on scale, functionality, and operational stability. • Exclude logging sensitive information! • Secure your logs. • Create meaningful alerts/notifications. • Focus on having insights on actionable and impactful results. • Train your analysts and enhance their skills. • Update/maintain your operation plans and components/assets as needed.

Intro to Log Analysis

Log analysis is an essential aspect of cyber security and system monitoring. At a high level, log analysis examines and interprets log event data generated by various sources (devices, applications, and systems) to monitor metrics and identify security incidents. It involves collecting, parsing, and processing log files to turn data into actionable objectives. By adopting an effective log analysis strategy, security teams can more accurately respond to security incidents and gain proactive insights into potential threats.

In this room, we will explore concepts related to log analysis methodology, effective logging practices, and common tools to aid detection and response.

Learning Objectives

- Learn log analysis best practices.
- Discover the essential tools for log analysis.
- Gain hands-on experience in analyzing logs by using multiple tools and technologies.

- Log Analysis Basics
    - Logs are pivotal in offering valuable insights into these systems' inner workings and interactions across the network.
    - What is a log: a stream of time-sequenced messages that record occurring events
    - What is log analysis: the process of making sense of the events captured in the logs to paint a clear picture of what has happened across the infrastructure.

- Logs are recorded events that can be related to application errors, system defaults, audited user actions, resource uses, network connections, etc.
- Each log entry contains relevant details to contextualize the event
    - Ex: Jul 28 17:45:02 10.10.0.4 FW-1: %WARNING% general: Unusual network activity detected from IP 10.10.0.15 to IP 203.0.113.25. Source Zone: Internal, Destination Zone: External, Application: web-browsing, Action: Alert.
    - This log entry signifies an event detected by a firewall regarding unusual network activity from an internal system, indicating a potential security concern. The relevant fields to consider in this example are:
    - `Jul 28 17:45:02` - This timestamp shows the event's date and time.
    - `10.10.0.4` - This refers to the system's IP address (the source) that generated the log.
    - `%WARNING%` - This indicates the severity of the log, in this case, **Warning**. Log entries are often given a severity level to categorize and communicate their relative importance or impact. These severity levels help prioritize responses, investigations, and actions based on the criticality of the events. Different systems might use slightly different severity levels, but commonly, you can expect to find the following increasing severity levels: Informational, Warning, Error, and Critical.
    - `Action: Alert` - In this case, the firewall's policy was configured to notify when such unusual activity occurs.
    - The remaining fields give us specific information related to the logged event. Specifically, that unusual network activity was detected *from* IP 10.10.0.15 *to* IP 203.0.113.25. Based on the `Source Zone` field, the traffic appears destined for the Internet (**External**), and the **Application** was categorized as **web-browsing**.

- Why are logs important?
    - System Troubleshooting: Analyzing system errors and waring logs helps IT teams to understand and quickly respond to system failures, minimizing downtime, and improving overall system reliability
    - Cyber Security Incidents: crucial in detecting and responding to security incidents, firewall logs, intrusion detection system logs, and system authentication logs contain vital info about potential threats and suspicious activity
    - Threat Hunting: use collected logs to actively search for advanced threats that may have evaded traditional security measures. Security analysts and threat hunters can analyze logs to look for unusual patterns, anomalies, and indicators of compromise that might indicate the presence of a threat actor
    - Compliance: log analysis ensures that organizations can provide accurate reports and demonstrate compliance with regulations such as GDPR, HIPAA, or PCI DSS
- Types of Logs
    - Application Logs: Messages from specific applications, providing insights into their status, errors, warnings and other operational details
    - Audit Logs: Events, actions, and changes occurring within a system or application, providing a history of user activities and system behavior
    - Security Logs: Security-related events like logins, permission alterations, firewall activities and other actions impacting system security
    - Server Logs: System Logs, event logs, error logs, and access logs, each offering distinct information about server operations

        - System Logs: Kernel activities, system errors, boot sequences, and hardware status, aiding in diagnosing system issues
        - Network Logs: Communication and activity within a network, capturing information about events, connections, and data transfers
        - Database logs: Activities within a database system, such as queries performed, actions and updates.
        - Web Server Logs: Requests processed by web servers, including URLs, source IP addresses, request types, response codes and more

- Investigation Theory
    - Timeline: creating a timeline is a fundamental aspect of understanding the sequence of events within systems, devices, and applications. At a high level, a timeline is a chronological representation of the logged events, ordered based on their occurrence.
    - Timestamp: logs will typically include timestamps that record when an event occurred.
        - With the potential of many distributed devices, applications, and systems generating individual log events across various regions, it's crucial to consider each log's time zone and format.
        - Converting timestamps to a consistent time zone is necessary for accurate log analysis and correlation across different log sources.
    - Super Timelines: also known as a consolidated timeline, is a powerful concept in log analysis and digital forensics. Super timelines provide a comprehensive view of events across different systems, devices, and applications, allowing analysts to understand the sequence of events holistically.

        - often include data from previously discussed log sources, such as system logs, application logs, network traffic logs, firewall logs, and more.
        - By combining these disparate sources into a single timeline, analysts can identify correlations and patterns that need to be apparent when analyzing logs individually.
        - Plaso (Python Log2Timeline) is an open-source tool created by Kristinn Gudjonsson and many contributors that automates the creation of timelines from various log sources. It's specifically designed for digital forensics and log analysis and can parse and process log data from a wide range of sources to create a unified, chronological timeline.
    - Data Visualization: Data visualization tools, such as Kibana (of the Elastic Stack) and Splunk, help to convert raw log data into interactive and insightful visual representations through a user interface.
        - Tools like these enable security analysts to understand the indexed data by visualizing patterns and anomalies, often in a graphical view
    - Log Monitoring and Alerting
        - implementing effective log monitoring and alerting allows security teams to *proactively* identify threats and immediately respond when an alert is generated.
        - Many SIEM solutions (like Splunk and the Elastic Stack) allow the creation of custom alerts based on metrics obtained in log events.
            - Events worth creating alerts for may include multiple failed login attempts, privilege escalation, access to sensitive files, or other indicators of potential security breaches.
- Common Log File Locations
    - Web Servers

        - Nginx
            - Access Logs: /var/log/nginx/access.log
            - Error Logs: /var/log/nginx/error.log
        - Apache
            - Access Logs: /var/log/apache2/access.log
            - Error Logs: /var/log/apache2/error.log
    - Databases
        - MySQL
            - Error Logs: /var/log/mysql/error.log
        - PostgreSQL
            - Error and Activity Logs: /var/log/postgresql/postgresql-{version}-main.log
    - Web Applications
        - PHP
            - Error Logs: /var/log/php/error.log
    - Operating System
        - Linux
            - General System Logs: /var/log/syslog
            - Authentication Logs: /var/log/auth.log
    - Firewalls and IDS/IPS
        - Iptables
            - Firewall Logs: /var/log/iptables.log
        - Snort

- Snort Logs: /var/log/snort

- Common Patterns
    - recognizing common patterns and trends in log data is crucial for identifying potential security threats
    - **Abnormal User Behavior:** unusual or anomalous user behavior
        - To effectively detect anomalous behavior, organizations can employ log analysis that incorporate detection engines and machine learning algorithms to establish normal behavior patterns
        - Deviations from these patterns or baselines can then be alerted as potential security incidents
    - Specific indicators can vary greatly depending on source, examples can be found in log files :
        - Multiple failed login attempts
        - Unusual login attempts
        - Geographical login times
        - Frequent password changes
        - Unusual user-agent strings
    - fine-tune any automated anomaly detection mechanisms to minimize false positives.
- Common Attack Signatures
    - SQL Injection
        - attempts to exploit vulnerabilities in web applications that interact with databases.
        - Look for unusual or malformed SQL queries in the application or database logs to identify common SQL injection patterns

        - Suspicious SQL queries might contain unexpected characters, such as single quotes (`'`), comments (`--`, `#`), union statements (`UNION`), or time-based attacks (`WAITFOR DELAY`, `SLEEP()`). A useful SQLi payload list to reference can be found here
            - Ex SQLI Log: 10.10.61.21 - - [2023-08-02 15:27:42] "GET /products.php?q=books' UNION SELECT null, null, username, password, null FROM users-- HTTP/1.1" 200 3122
    - Cross-Site Scripting (XSS)
        - Exploiting cross-site scripting (XSS) vulnerabilities allow attackers to inject malicious scripts into web pages
        - look for log entries with unexpected or unusual input that includes script tags (`<script>`) and event handlers (`onmouseover`, `onclick`, `onerror`)
        - useful XSS payload list to reference can be found here.
        - Ex xss log: 10.10.19.31 - - [2023-08-04 16:12:11] "GET /products.php?search=<script>alert(1);</script> HTTP/1.1" 200 5153
    - Path Traversal
        - Exploiting path traversal vulnerabilities allows attackers to access files and directories outside a web application's intended directory structure, leading to unauthorized access to sensitive files or code.
        - ook for traversal sequence characters (`../` and `../../`) and indications of access to sensitive files (`/etc/passwd`, `/etc/shadow`). A useful directory traversal payload list to reference can be found here.
        - directory traversals are often URL encoded (or double URL encoded) to avoid detection by firewalls or monitoring tools. Because of this, `%2E` and `%2F` are useful URL-encoded characters to know as they refer to the `.` and `/` respectively

        - Ex path-traversal log: 10.10.113.45 - - [2023-08-05 18:17:25] "GET /../../../../../etc/passwd HTTP/1.1" 200 505

- Automated vs Manual Analysis
    - Automated Analysis
        - allow for processing and data analysis of logs. These tools often utilize Artificial Intelligence / Machine Learning to analyze patterns and trends

Advantages	**Disadvantages**
Saves time by performing a lot of the manual work required in manual analysis	Automated analysis tools are usually commercial-only and, therefore, expensive.
The use of artificial intelligence is effective at recognizing patterns and trends.	The effectiveness of artificial intelligence depends on how capable the model is. For example, the risk of false positives increases, or newer or never-seen-before events can be missed as the AI is not trained to recognize these.

    - Manual Analysis
        - Manual analysis is the process of examining data and artifacts without using automation tools.

Advantages	**Disadvantages**
It is cheap and does not require expensive tooling. For example, simple Linux commands can do the trick.	It is time-consuming as the analyst has to do all of the work, including reformatting log files.
Allows for a thorough investigation.	N/A
Reduces the risk of overfitting or false positives on alerts from automated tools.	Events or alerts can be missed! Especially if there is a lot of data to comb through.

Allows for contextual analysis. The analyst has a broader understanding of the organization and cyber security landscape.	N/A

- Log Analysis Tools: Command Line
    - The `less` command is an improvement over `cat` when dealing with larger files. It allows you to view the file's data page by page, providing a more convenient way to read through lengthy logs
    - When using `less` to open a file, it displays the first page by default, and you can scroll down using the arrow keys or with *Page Up* and *Page Down*.
    - The `tail` command is specifically designed for viewing the end of files and is very useful for seeing a summary of recently generated events in the case of log files.
    - The most common use of `tail` is coupled with the `-f` option, which allows you to "follow" the log file in real-time, as it continuously updates the terminal with new log entries as they are generated and written. This is extremely useful when monitoring logs for live events or real-time system behavior.
    - The `wc` (word count) command is a simple but powerful utility that can be quite useful for quick analysis and statistics gathering.
    - The `cut` command extracts specific columns (fields) from files based on specified delimiters. This is a handy command for working with log files that have structured or tab-separated data.
    - The `sort` command arranges the data in files in ascending or descending order based on specific criteria
    - The `uniq` command identifies and removes adjacent duplicate lines from sorted input

    - We can also append the `-c` option to output unique lines and prepend the count of occurrences for each line. This can be very useful for quickly determining IP addresses with unusually high traffic.
    - Both `sed` and `awk` are powerful text-processing tools commonly used for log analysis. They are sometimes used interchangeably, but both commands have their use cases and can allow security analysts to manipulate, extract, and transform log data efficiently.
    - Using the substitute syntax, `sed` can replace specific patterns or strings into log entries.
    - Note that the backslash character `\` is required to "escape" the forward slash in our pattern and tell `sed` to treat the forward slash as a literal character.
    - `sed` command *does not* change the `apache.log` file directly; instead, it only outputs the modified version of the file to the standard output in the command line. If you want to overwrite the file, you can add the `-i` option to edit the file in place or use a redirect operator `>` to save the output to the original or another file.
    - For the `awk` command, a common use case is conditional actions based on specific field values.
    - The `grep` command is a powerful text search tool widely used on UNIX systems and provides exceptional use cases in log analysis.
    - allows you to search for specific patterns or regular expressions within files or streams of text. Using `grep` can help analysts quickly identify relevant log entries that match specific criteria, particular resources or keywords, or patterns
    - The most basic usage of `grep` is to search for specific strings within log files.

- Log Analysis Tools: Regular Expressions

- Regular expression patterns are constructed using a combination of special characters that represent matching rules and are supported in many programming languages, text editors, and software.

Splunk: Exploring SPL

-