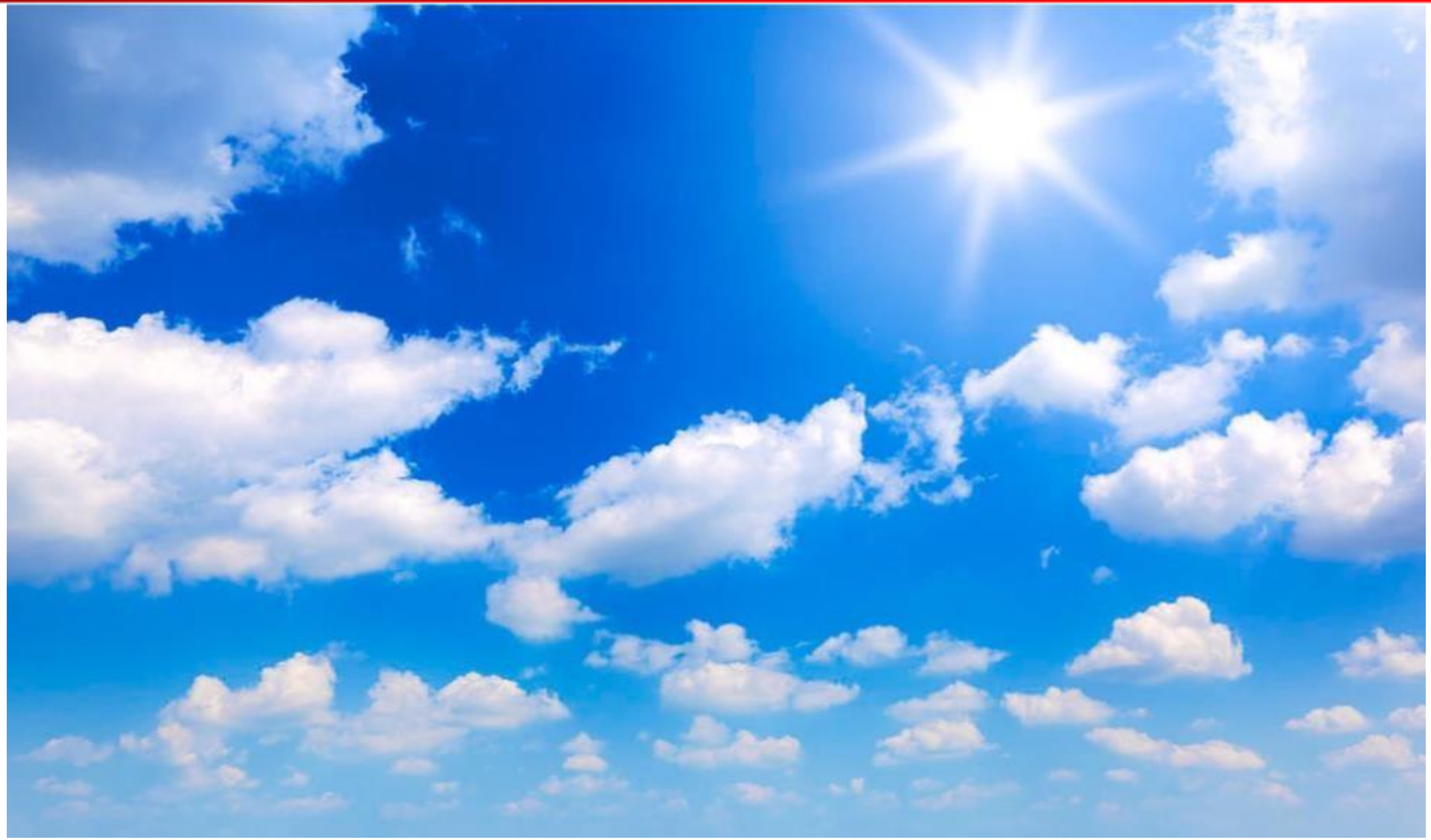
# Cyber Security
# Lession 6



# In the previous lession…

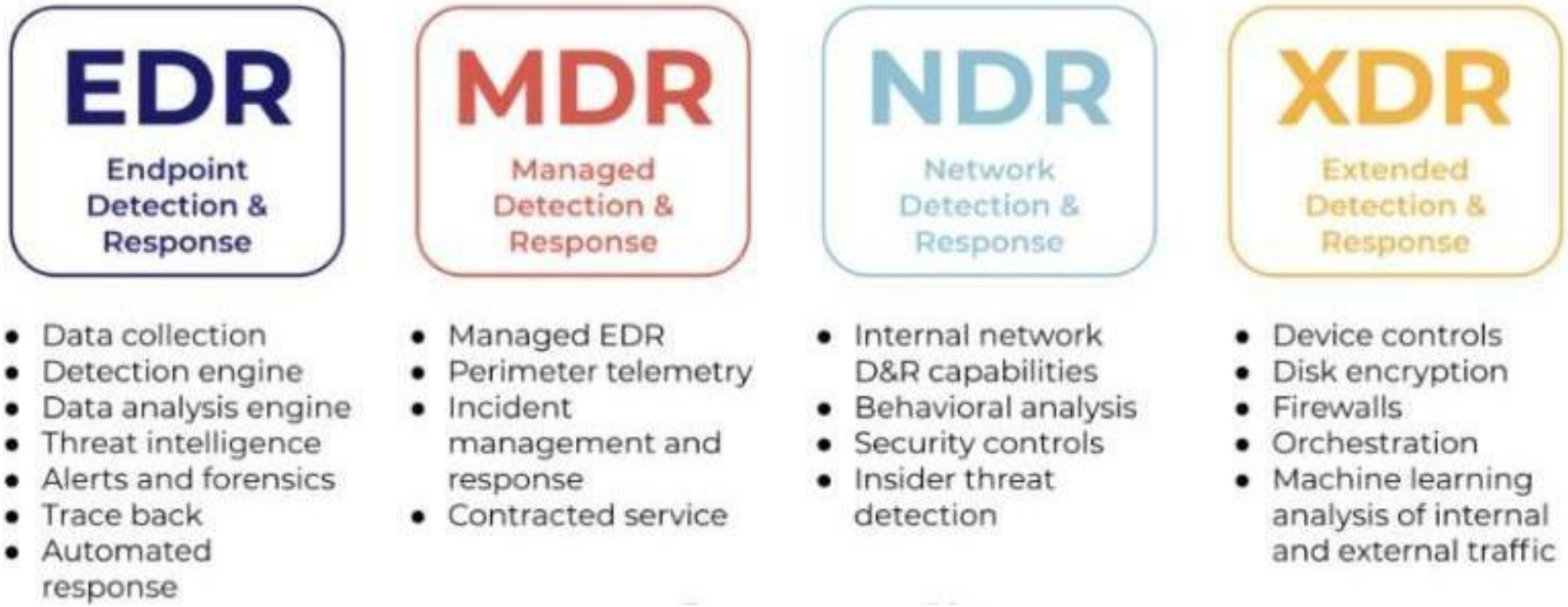
# How Advanced Ransomware malware work

- Exploited vulnerability
- Phishing
- Compromised credentials
- Brute-force attacks
- Misconfigured service
- Malicious attachments / downloads

# How Advanced Ransomware malware work

1. Disables factory OS protection.
2. Data Exfiltration (sometimes).
3. Computer / server backups being removed or corrupted.
4. Ransomware encrypting files.
5. Ransomware removes OS log files.
6. Ransomware note created.
7. Ransomware erases its core process from the system.

# Detection & Responses Types



# Introduction

- Understanding Core Security Goals DONE
- Introducing Basic Risk Concepts DONE
- Understanding Security Controls DONE
- Using Command-Line Tools DONE
- Understanding Logs

# Understanding Logs

# Logs

- Entries help administrators and security investigators determine what happened, when it happened, where it happened, and who or what did it.
- When examining entries from multiple logs, personnel create an audit trail that identifies all the events preceding a security incident.

# Windows logs

- Logs are viewable using the Windows Event Viewer.
- The primary Windows logs are:
  - ✓ **Security log.** The Security log functions as a security log, an audit log, and an access log. It records auditable events such as successes or failures. Success indicates an audited event completed successfully, such as a user logging on or successfully deleting a file. Failure means that a user tried to perform an action but failed, such as failing to log on or attempting to delete a file but receiving a permission error instead. Windows enables some auditing by default, but administrators can add additional auditing.
  - ✓ **System.** The operating system uses the System log to record events related to the functioning of the operating system. This can include when it starts, when it shuts down, information on services starting and stopping, drivers loading or failing, or any other system component event deemed important by the system developers.
  - ✓ **Application log**. The Application log records events sent to it by applications or programs running on the system. Any application has the capability of writing events in the Application log. This includes warnings, errors, and routine messages.

# Network logs

- Network logs record traffic on the network.
- These logs are on a variety of devices such as routers, firewalls, web servers, and network intrusion detection/prevention systems.
- You can typically manipulate devices to log specific information, such as logging all traffic that the device passes, all traffic that the device blocks, or both.
- These logs are useful when troubleshooting connectivity issues and when identifying potential intrusions or attacks.
- They include information on where the packet came from (the source) and where it is going (the destination). This includes IP addresses, MAC addresses, and ports.
- Web servers typically log requests to the web server for pages.
- These often follow the Common Log format standardized by the World Wide Web Consortium (W3C). A typical entry includes the following data:
  - ✓ **host:** The IP address or hostname of the client requesting the page.
  - ✓ **user-identifier:** The name of the user requesting the page (if known)
  - ✓ **authuser**: The logon name of the user requested in the page, if the user logged on.
  - ✓ **date**: The date and time of the request.
  - ✓ **request**: The actual request line sent by the Ćlient.
  - ✓ **status**: The HTTP status code returned to the client bytes: The byte length of the reply.

# Wevtutil

- **Windows Events Command Line Utility** an administrator command line utility used primarily to register your event provider on the computer.
- Provides metadata information about the provider, its events, and the channels to which it logs events, and to query events from a channel or log file.
- Install and uninstall event manifests.
- Run queries.
- Export logs.
- Archive logs.
- Clear logs.

# Centralized Logging Methods

- It can be quite challenging to routinely check the logs on all the devices within a network.
- A standard solution is to use a centralized system to collect log entries.
- Two popular methods are:
  - ✓SIEM system;
  - ✓syslog protocol.

# SIEM

- **A security information and event management** (SIEM) system provides a centralized solution for collecting, analyzing, and managing data from multiple sources.
- It combines the services of security event management (SEM) and security information management (SIM) solutions.
- A SEM provides real-time monitoring, analysis, and notification of security events, such as suspected security incidents.
- A SIM provides long-term storage of data, along with methods of analyzing the data looking for trends or creating reports needed to verify compliance with laws or regulations.
- A benefit is that SIEM systems use scripts to automate the monitoring and reporting.
- Vendors sell SIEMs as applications that can be installed on centralized systems and as dedicated hardware appliances. However, no matter how a vendor bundles it, it will typically have common capabilities.

# Additional capabilities shared by most SIEMs

- **Log collectors.** The SIEM collects log data from devices throughout the network and stores these logs in a searchable database.
- **Data inputs.** Log entries come from various sources, such as firewalls, routers, network intrusion detection and prevention systems, and more. They can also come from any system that an organization wants to monitor, such as web servers, proxy servers, and database servers.
- **Log aggregation.** Aggregation refers to combining several dissimilar items into a single similar format. The SIEM system collects data from multiple systems, and these systems typically format log entries differently. However, the SIEM system can aggregate the data and store it so that it is easy to analyze and search.
- **Correlation engine.** A correlation engine is a software component used to collect and analyze event log data from various systems within the network. It typically aggregates the data looking for common attributes. It then uses advanced analytic tools to detect patterns of potential security events and raises alerts. System administrators can then investigate the alert.
- **Reports.** Most SIEM systems include multiple built-in reports. These are typically grouped in different categories such as network traffic event monitoring, device events (such as events on border firewalls), threat events, logon/logoff events, compliance with specific laws, and more. Additionally, security professionals can create their own reports by specifying filters. Packet capture. Protocol analyzers (sometimes called sniffers) capture network traffic allowing administrators to view and analyze individual packets. However, many SIEM systems include the same capabilities.
- **User behavior analysis.** User behavior analysis (**UBA**) focuses on what users are doing, such as what applications they are launching and their network activity. Some UBA processes watch critical files looking for who accessed them, what they did, and how frequently they access these files. UBA typically looks for abnormal patterns of activity that may indicate malicious intent. Some **data loss prevention** (**DLP**) systems include this ability.
- **Sentiment analysis.** Generically, sentiment analysis refers to analyzing text to detect an opinion or emotion. Within a SIEM system, it refers to using UBA technologies to observe user behaviors to detect unwanted behavior. This is no small feat and typically relies on artificial intelligence to analyze large data sets.
- **Security monitoring.** A SIEM typically comes with predefined alerts, which can provide continuous monitoring of systems and provide notifications of suspicious events. SIEMs also include the ability to create new alerts. Automated triggers. Triggers cause an action in response to a predefined number of repeated events. A SIEM includes the ability to modify predefined triggers and create new ones.
- **Time synchronization.** All servers sending data to the SIEM should be synchronized with the same time. This becomes especially important when investigating an incident so that security investigators know when events occurred. Additionally, large organizations can have locations in different time zones. Each of these locations might have servers sending data to a single centralized SIEM. If the server logs use their local time, the SIEM needs to ensure that it compensates for the time offset. One method is to convert all times to **Greenwich Mean Time** (**GMT**), which is the time at the Royal Observatory in Greenwich, London.
- **Event deduplication.** Deduplication is the process of removing duplicate entries.
- **Logs/WORM.** A SIEM typically includes methods to prevent anyone from modifying log entries. This is sometimes referred to as **write once read many** (WORM). As logs are received, the SIEM aggregates and correlates the log entries. After processing the logs, it can archive the source logs with write protection.

# The location of the SIEM

- Varies based on how the SIEM is used.
- It's common to locate the SIEM within the private network, even if it collects data from a screened subnet.
- The internal network provides the best protection for the log data.
- In very large organizations, aggregation processes and the correlation engine can consume a lot of processing power, so organizations sometimes off-load these processes to another server (the primary SIEM appliance can then focus on alerts and triggers).

# SIEM dashboard

- Gives administrators views of meaningful activity.
- These views vary depending on the application developer and are usually customizable, but they provide continuous monitoring and real-time reporting.
- In a large **network operations center** (**NOC**), the SIEM might display alerts on a large heads-up display. In a smaller network, a single computer may show the dashboard.
- Common elements of a SIEM dashboard:
  - ✓ **Sensors**. Many SIEM systems use agents placed on systems throughout a network. These collect logs from devices and send these logs to the SIEM system. Dashboards can display data received from these agents.
  - ✓ **Alerts**. After setting triggers in a SIEM system, it sends out alerts when the event fires. These alerts may trigger specific responses (such as sending an email to a group), but they are also displayed in the dashboard.
  - ✓ **Sensitivity**. A challenge with triggers and alerts is setting the sensitivity levels to limit false positives while avoiding false negatives.
  - ✓ **Correlation**. As log entries arrive at the SIEM system, it correlates and analyzes the data. Administrators can configure the dashboard to display this data in multiple ways depending on their needs.
  - ✓ **Trends**. As the SIEM system is analyzing the data, it can identify trends. Many SIEM systems display trends in graphs allowing users to digest a lot of information in a single picture.

# Syslog protocol

- Specifies a general log entry format and the details on how to transport log entries.
- You can deploy a centralized syslog server to collect syslog entries from a variety of devices in the network, similar to how a SIEM server collects log entries.
- Syslog was developed in the 1980s and became a standard on Unix- like systems. However, there wasn't a single publication that defined it. **The Institute of Electrical and Electronics Engineers** (**IEEE**) documented it in an informational a **Request for Comments** (**RFC**) RFC 3164 in 2001. In 2009, they documented it as a standard in RFC 5424.
- Any **systems sending** syslog messages are **originators**.
- **Originators send** syslog log entries **to a collector** (a syslog server).
- The **collector** can **receive** messages from external devices or services and applications on the same system.
- The syslog **protocol** only **defines how to format** the syslog messages and send them to a collector. However, it **doesn't define how** the syslog server **handles** these log entries.
- **Often Linux** systems **include** the syslogd daemon **or** you must install service using **apt-get**, which is the service that handles the syslog messages. It collects the entries and processes them **based on entries** in the **/etc/syslog.conf file (rsyslog.conf)**. Many syslog messages are **routed** to the **/var/syslog** file.
- It's also possible to use additional applications to collect and process syslog entries. Some sophisticated applications using syslog can perform many of the same functions of a SIEM system.
- Historically, systems sent syslog messages via UDP using port 514.
- UDP doesn't provide guaranteed delivery. Newer implementations can use TCP port 6514 with Transport Layer Security (TLS). TCP ensures the packets arrive, and TLS provides encryption.

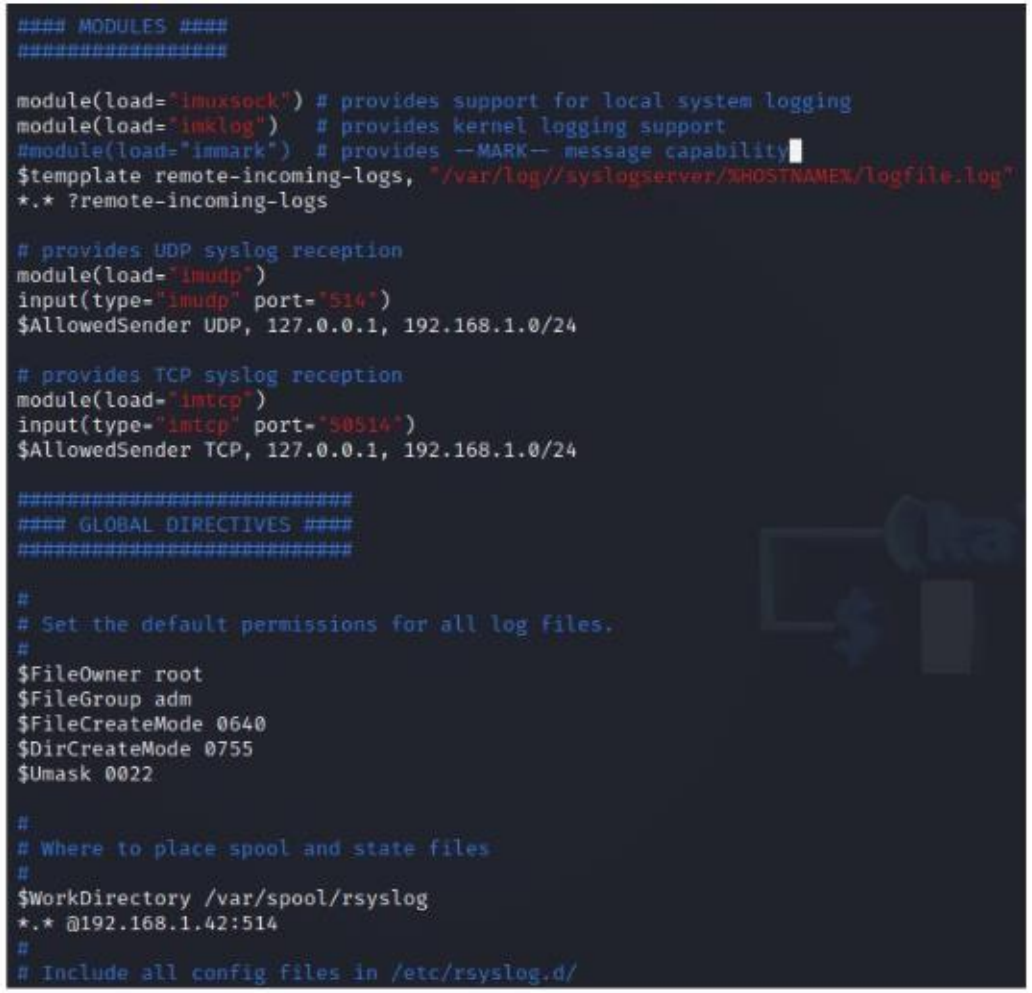
# Syslog-ng and Rsyslog

- Additional open source software utilities are used instead of syslogd on Linux-like systems.
- These are based on syslogd but provide additional extensions.
- **Syslog-ng**. Syslog-ng extends syslogd, allowing a system to collect logs from any source. It also includes correlation and routing abilities to route log entries to any log analysis tool. It provides rich filtering capabilities, content-based filtering, and can be extended with tools and modules written in other languages. It supports TCP and TLS.
- **Rsyslog**. Rsyslog came out later as an improvement over syslog-ng. One significant change is the ability to send log entries directly into database engines. It also supports TCP and TLS.

# KALI Rsyslog

1) ifconfig (sudo dhclient -r, sudo dhclient)
2) sudo apt-get install rsyslog -y
3) sudo systemctl enable --now rsyslog
4) sudo cp /etc/rsyslog.conf /etc/rsyslog.conf.backup
5) sudo vim /etc/rsyslog.conf (Shift+I - Esc, :w, :x )
6) sudo systemctl enable --now rsyslog
7) sudo apt install ufw
8) sudo ufw allow 514/udp
9) sudo ufw allow 50514/tcp
10) sudo ufw allow from 192.168.1.0/24 to any port 514 proto udp
11) sudo ufw allow from 192.168.1.0/24 to any port 50514 proto tcp
12) sudo systemctl restart ufw
13) sudo systemcl restart rsyslog

[No Title]



# NXLog

- NXLog is another log management tool and is similar to rsyslog and syslog-ng.
- Supports log formats for Windows, such as event log entries.
- It can be installed on both Windows and Linux-like systems.
- It functions as a log collector, and it can integrate with most SIEM systems.
- It comes in two versions:
  - ✓ **NXLog Community Edition**. The Community Edition is a propriety log management tool available from https://nxlog.co. Installation packages are available for Microsoft Windows and GNU/Linux. While it's free, it includes a feature set comparable with some SIEM solutions.
  - ✓ **NXLog Enterprise Edition**. The Enterprise Edition includes all the features of the Community Edition but adds additional capabilities. It provides real-time event correlation and remote administration.

# Linux Logs

- **var/log/syslog**. The syslog file stores all system activity, including startup activity. This is not the syslog protocol used to collect log entries from other systems.
- **var/log/messages**. This log contains a wide variety of general system messages. It includes some messages logged during startup, some messages related to mail, the kernel, and messages related to authentication.
- **var/log/boot.log**. This log includes entries created when the system boots.
- **var/log/auth.log**. The authentication log contains information related to successful and unsuccessful logins.
- **var/log/faillog**. This log contains information on failed login attempts. It can be viewed using the faillog command.
- **/var/log/kern.log**. The kernel log contains information logged by the system kernel, which is the central part of the Linux operating system.
- **/var/log/httpd/.** If the system is configured as an Apache web server, you can view access and error logs within this directory.

# Journalctl

- Some of the most compelling advantages of systemd are those involved with process and system logging.
- When using other tools, logs are usually dispersed throughout the system, handled by different daemons and processes, and can be fairly difficult to interpret when they span multiple applications.
- systemd attempts to address these issues by providing a centralized management solution for logging all kernel and userland processes.
- The **system that collects and manages these logs** is known as the **journal**.
- The journal is implemented with the **journald daemon**, which handles all of the messages produced by the kernel, initrd, services, etc.

# Journal general idea

- One of the impetuses behind the systemd journal is to centralize the management of logs regardless of where the messages are originating.
- Since much of the boot process and service management is handled by the systemd process, it makes sense to standardize the way that logs are collected and accessed.
- The journald daemon collects data from all available sources and stores them in a binary format for easy and dynamic manipulation.
- By interacting with the data using a single utility, administrators are able to dynamically display log data according to their needs.
- Storing the log data in a binary format also means that the data can be displayed in arbitrary output formats depending on what you need at the moment.
- For instance, for daily log management you may be used to viewing the logs in the standard syslog format, but if you decide to graph service interruptions later on, you can output each entry as a JSON object to make it consumable to your graphing service.
- Since the data is not written to disk in plain text, no conversion is needed when you need a different on-demand format.
- The systemd journal can either be used with an existing syslog implementation, or it can replace the syslog functionality, depending on your needs.
- While the systemd journal will cover most administrator's logging needs, it can also complement existing logging mechanisms.
- For instance, you may have a centralized syslog server that you use to compile data from multiple servers, but you also may wish to interleave the logs from multiple services on a single system with the systemd journal. You can do both of these by combining these technologies.

# Journal main commands

- journalctl --utc
- journalctl -b
- journalctl -b -1
- sudo nano /etc/systemd/journald.conf
- journalctl --since "2015-01-10 17:15:00"
- journalctl --since "2015-01-10" --until "2015-01-11 03:00"
- journalctl --since yesterday
- journalctl --since 09:00 --until "1 hour ago"
- journalctl -u nginx.service
- journalctl _UID=33 --since today
- journalctl /usr/bin/bash
- journalctl -p err -b
- journalctl --no-pager
- journalctl --disk-usage
- sudo journalctl --vacuum-size=1G
- journalctl -f Realtime
- journalctl -b -u nginx -o json
- journalctl -b -u nginx -o json-pretty

- Formats :
  - ✓ **cat**: Displays only the message field itself.
  - ✓ **export**: A binary format suitable for transferring or backing up.
  - ✓ **json**: Standard JSON with one entry per line.
  - ✓ **json-pretty**: JSON formatted for better human-readability
  - ✓ **json-sse**: JSON formatted output wrapped to make add server-sent event compatible
  - ✓ **short**: The default syslog style output
  - ✓ **short-iso**: The default format augmented to show ISO 8601 wall clock timestamps.
  - ✓ **short-monotonic**: The default format with monotonic timestamps.
  - ✓ **short-precise**: The default format with microsecond precision
  - ✓ **verbose**: Shows every journal field available for the entry, including those usually hidden internally.

# Impact of Log File Deletion

- **Compliance Obligations and Auditing**: Log files are vital for meeting compliance obligations and facilitating auditing processes. Before proceeding with log file deletion, it is essential to carefully review relevant compliance regulations and retention policies to ensure adherence to data retention requirements.
- **Incident Response and Digital Forensics**: Log files serve as valuable sources of information for incident response and digital forensics investigations. Premature deletion of log files can hinder the ability to identify the root cause of security incidents, track malicious activities, and establish a comprehensive timeline of events. Evaluate the potential impact on incident response capabilities and forensic analysis.
- **Troubleshooting and Debugging**: Log files are commonly relied upon for troubleshooting and debugging purposes. They offer insights into system behavior, error messages, and warnings that aid in diagnosing and resolving issues. Deleting log files prematurely may impede effective investigation and resolution of future system or application problems.
- **Performance Monitoring and Analysis**: Log files play a crucial role in log performance monitoring and analysis, enabling the identification of performance bottlenecks, resource utilization tracking, and system performance optimization. Deleting log files may hinder the ability to conduct historical performance analysis and identify long-term trends for informed decision-making.
- **Compliance Reporting**: Log files contribute to compliance reporting efforts, demonstrating adherence to security standards and regulations. Premature deletion of log files can negatively impact the ability to generate accurate compliance reports and demonstrate compliance during audits.
- Storage Management and Disk Space: Over time, log files can consume substantial storage space. Deleting log files helps alleviate storage capacity issues and enhances overall system performance. Consider the implications on storage management strategies and ensure alignment with disk space optimization goals.
- **Risk Assessment**: Perform a comprehensive risk assessment to evaluate the potential impact of log file deletion within the specific context of your organization. Consider industry requirements, business needs, and security considerations to strike a balance between data retention requirements and the potential benefits gained from deleting log files.

# Windows log clear

**CMD**

1) Open an elevated command prompt.
2) Type or paste the following command:

```
for /F "tokens=*" %1 in ('wevtutil.exe el') DO wevtutil.exe cl "%1"
```

**PowerShell**

1) Open PowerShell as administrator.
2) Type or copy-paste the following command into PowerShell:

```
wevtutil el | Foreach-Object {wevtutil cl "$_"}
```

# Clear logs on Linux with Journal

1) sudo journalctl --flush --rotate
2) sudo journalctl --vacuum-time=1s