TryHackMe Journal - Rowan King

Instructions

(1) Review the sample journal entry provided below
(2) Scroll down to find the name of the room you have been assigned/are working on
(Pro Tip: Turn on "Outline View" so you can navigate more easily - go to View → Show Outline)
(3) Complete the required rooms on TryHackMe, compiling notes as you work through the room. This might include:
  (a) Commonly used Code/Commands
  (b) Definitions/Explanations of important terms and concepts
  (c) Screenshots of useful diagrams
(4) Once you've completed the module, capture 2-4 important takeaways.
(5) After you get the hang of things, delete these instructions and the sample you were provided!

Entry 1

Room Name: Linux Fundamentals 1

Date Completed: 4/27/2024
Notes During the Room: The basics of Linux and the CLI

Introduced several commands;

Command	**Description**
echo	Output any text that we provide
whoami	Find out what user we're currently logged in as!

ls	listing
cd	change directory
cat	concatenate
pwd	print working directory
find	search for a specified file or folder
grep	search the contents of files for specific values that we are looking for

As well as Operators:

Symbol / Operator	Description
&	This operator allows you to run commands in the background of your terminal.
&&	This operator allows you to combine multiple commands together in one line of your terminal.
>	This operator is a redirector - meaning that we can take the output from a command (such as using cat to output a file) and direct it elsewhere.
>>	This operator does the same function of the `>` operator but appends the output rather than replacing (meaning nothing is overwritten).

Important Takeaways:

Entry 2

Room Name: Linux Fundamentals 2

Date Completed: 4/27/2024
Notes During the Room:
Introduced remote login using the SSH command

SSH *	The syntax to use SSH is very simple. We only need to provide two things: 1. The IP address of the remote machine

	2. Correct credentials to a valid account to login with on the remote machine

Other Filesystem Commands

Command	Full Name	Purpose
touch	touch	Create file
mkdir	make directory	Create a folder
cp	copy	Copy a file or folder
mv	move	Move a file or folder
rm	remove	Remove a file or folder
file	file	Determine the type of a file

Covered common directories such as /ect, /var, /root, & /tmp

Important Takeaways:

Entry 3

Room Name: Linux Fundamentals 3

Date Completed: 4/27/2024
Notes During the Room:

Learned about nano and vim.

Learned about wget and its variables;

Variable	Value
The IP address of the remote system	192.168.1.30
User on the remote system	ubuntu
Name of the file on the local system	important.txt
Name that we wish to store the file as on the remote system	transferred.txt

Learned about running and killing processes

Learned about cron files used to automate tasks;

Value	Description
MIN	What minute to execute at
HOUR	What hour to execute at
DOM	What day of the month to execute at
MON	What month of the year to execute at
DOW	What day of the week to execute at
CMD	The actual command that will be executed.

Learned about adding and removing repositories

Learned how to pull system logs

Important Takeaways:

Entry 4

Room Name: Linux Strength Training

Date Completed:
Notes During the Room:
As a security researcher you will often be required to find specific files/folders on a system based on various conditions ranging from, but not limited to the following:

- **filename**
- **size**
- **user/group**
- **date modified**
- **date accessed**
- **Its keyword contents**

Therefore, we can do this using the following syntax:

What we can do	Syntax	Real example of syntax
Find files based on filename	find [directory path] -type f -name [filename]	find /home/Andy -type f -name sales.txt
Find Directory based on directory name	find [directory path] -type d -name [filename]	find /home/Andy -type d -name pictures

Find files based on size	find [directory path] -type f -size [size]	find /home/Andy -type f -size 10c (c for bytes, k for kilobytes M megabytes G for gigabytes type:'man find' for full information on the options)
Find files based on username	find [directory path] -type f -user [username]	find /etc/server -type f -user john
Find files based on group name	find [directory path] -type f -group [group name]	find /etc/server -type f -group teamstar

Find files modified after a specific date	find [directory path] -type f -newermt '[date and time]'	find / -type f -newermt '6/30/2020 0:00:00' (all dates/times after 6/30/2020 0:00:00 will be considered a condition to look for)
Find files based on date modified	find [directory path] -type f -newermt [start date range] ! -newermt [end date range]	find / -type f -newermt 2013-09-12 ! -newermt 2013-09-14 (all dates before 2013-09-12 will be excluded; all dates after 2013-09-14 will be excluded, therefore this only leaves 2013-09-13 as the date to look for.)

Find files based on date accessed	find [directory path] -type f -newerat [start date range] ! -newerat [end date range]	find / -type f -newerat 2017-09-12 ! -newerat 2017-09-14 (all dates before 2017-09-12 will be excluded; all dates after 2017-09-14 will be excluded, therefore this only leaves 2017-09-13 as the date to look for.)
Find files with a specific keyword	grep -iRl [directory path/keyword]	grep -iRl '/folderA/flag'
read the manual for the find command	man find	man find

Note: There are many more useful commands aside from the examples above. If you ever have trouble understanding any of the syntax or getting it to work, head on over to explainshell.com to check the syntax and see how this tool can help you on your journey to Linux greatness.

Further notes: if you do not know already, typing CTRL+L allows you to clear the screen quicker rather than typing 'clear' all the time. Additionally, hitting the up arrow allows you to return to a previously typed command so you do not have to spend time retyping it again if you made an error. Cool. Finally, placing: **2>/dev/null** at the end of your find command can help filter your results to exclude files/directories that you do not have permission to.

You should be somewhat familiar already with working with files. Similar to windows, we can do the following:

- **copy files and folders**
- **move files and folders**
- **rename files and folders**
- **create files and folders**

For a quick recap to train your mental memory on the commands please refer to the below information:

What we can do	Syntax	Real example of syntax

copy file/folder	cp [filename/folder] [directory] (remember, if the filename/folder name has spaces then you will need to encase the filename with speech marks such as cp "[filename with spaces]" [directory]. This applis to other commands such as mv.)	cp ssh.conf /home/newfolder

move file/folder	mv [filename] [directory]	mv ssh.conf /home/newfolder
move multiple files/folders simultaneously	mv [file1] [file2] [file3] -t [directory to move to]	mv logs.txt keys.conf script.py -t /home/savedWork
Move all files from current directory into another directory	mv * [directory to move files to]	mv * /home/scripts
rename files/folder	mv [current filename] [new filename]	mv ssh.conf NewSSH.conf
create a file	touch [filename]	touch newFile.txt
create a folder	mkdir [foldername]	mkdir newFolder
open file for editing	nano [filename]	nano keys.conf
output contents of file	cat [filename]	cat keys.conf
upload file to a remote machine	scp [filename] [username]@[IP of remote machine]:/[directory to upload to]	scp example.txt john@192.168.100.123:/home/john/
run an bash script program	./[name of script]	./timer
open a file for reading/editing	nano [filename]	nano readME.txt

A few additional things to remember is that occasionally you may encounter files/folders with special characters such as - (dash). Just remember that if you try to copy or move these files you will encounter errors because Linux interprets the - as a type of argument, therefore you will have to place -- just before the filename. For example: cp -- -filename.txt /home/folderExample.

Important Takeaways:

Entry 5

Room Name: Intro to Logs

Date Completed: 5/26/2024
Logs serve as invaluable records of past events, providing essential insights to address these questions. By preserving an archive of historical activities, we can bolster our security posture and protect our digital assets more effectively.

Scenario: A web server of **SwiftSpend Financial** is constantly bombarded with scans from an adversary. As a systems administrator of this organisation tasked to address this predicament, you must identify what the adversary is doing by configuring logging and analysing collected logs.

Just as a physical tree's rings reveal its life story – indicating good years with thick curls and challenging ones with thin – a digital log provides a historical record of system activity.

Both embody a fundamental principle of growth over time and serve as living records in their respective domains – physical and digital.

In the digital world, every interaction with a computer system – from authentication attempts, granting authorisation, accessing a file, and connecting to a network to encountering a system error – will always leave a digital footprint in the form of logs.

Logs are a record of events within a system. These records provide a detailed account of what a system has been doing, capturing a wide range of events such as user logins, file accesses, system errors, network connections, and changes to data or system configurations.

While the specific details may differ based on the type of log, a log entry usually includes the following information:

- A timestamp of when an event was logged
- The name of the system or application that generated the log entry
- The type of event that occurred
- Additional details about the event, such as the user who initiated the event or the device's IP address that generated the event

This information is typically stored in a **log file**, which contains aggregated entries of what occurred at any given time on a system.

However, since digital interactions are continuous and fast-paced, the log file's size may exponentially grow depending on the activities logged on a system.

The True Power of Logs: Contextual Correlation

A single log entry may seem insignificant on its own. But when log data is aggregated, analysed, and cross-referenced with other sources of information, it becomes a potent investigation tool. Logs can answer critical questions about an event, such as:

- **What** happened?
- **When** did it happen?
- **Where** did it happen?
- **Who** is responsible?
- **Were** their actions **successful**?
- **What** was the result of their action?

The following hypothetical scenario can illustrate this aspect. Suppose a student allegedly accessed inappropriate content on a University network. By reviewing the logs, a systems administrator could then answer the following:

Question	Answer
What happened?	An adversary was confirmed to have accessed SwiftSpend Financial's GitLab instance.
When did it happen?	Access started at 22:10 on Wednesday, September 8th, 2023.
Where did it happen?	The event originated from a device with an IP address of 10.10.133.168 within the VPN Users' segment (10.10.133.0/24).
Who is responsible?	Upon examining the network logs, it was observed that the device, identified by the User-Agent "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0", was allocated the IP address 10.10.133.168.

Were they successful?	Yes, since an API Key was found to be publicly exposed on the GitLab instance. Moreover, the web proxy logs confirm that the adversary device reached *gitlab.swiftspend.finance* and maintained access through their uploaded web shell.
What is the result of their action?	The adversary achieved remote code execution on *gitlab.swiftspend.finance* and performed post-exploitation activities.

The example above emphasizes how logs are instrumental in piecing together a complete picture of an event, thereby enhancing our understanding and ability to respond effectively.

Log Types

Specific log types can offer a unique perspective on a system's operation, performance, and security. While there are various log types, we will focus on the most common ones that cover approximately 80% of the typical use cases.

Below is a list of some of the most common log types

- **Application Logs:** Messages about specific applications, including status, errors, warnings, etc.
- **Audit Logs:** Activities related to operational procedures crucial for regulatory compliance.
- **Security Logs:** Security events such as logins, permissions changes, firewall activity, etc.
- **Server Logs:** Various logs a server generates, including system, event, error, and access logs.
- **System Logs:** Kernel activities, system errors, boot sequences, and hardware status.
- **Network Logs:** Network traffic, connections, and other network-related events.
- **Database Logs:** Activities within a database system, such as queries and updates.
- **Web Server Logs:** Requests processed by a web server, including URLs, response codes, etc.

Understanding the various log types, formats, and standards is critical for practical log analysis. It enables an analyst to effectively parse, interpret, and gain insights from log data, facilitating troubleshooting, performance optimization, incident response, and threat hunting.

Log Formats

A log format defines the structure and organization of data within a log file. It specifies how the data is encoded, how each entry is delimited, and what fields are included in each row. These formats can vary widely and may fall into three main categories: Semi-structured, Structured, and Unstructured.

Log Standards

A log standard is a set of guidelines or specifications that define how logs should be generated, transmitted, and stored. Log standards may specify the use of particular log formats, but they also cover other aspects of logging, such as what events should be logged, how logs should be transmitted securely, and how long logs should be retained. Examples of log standards include:

> **Common Event Expression (CEE):** This standard, developed by MITRE, provides a common structure for log data, making it easier to generate, transmit, store, and analyse logs.

- **OWASP Logging Cheat Sheet:** A guide for developers on building application logging mechanisms, especially related to security logging.
- **Syslog Protocol:** Syslog is a standard for message logging, allowing separation of the software that generates messages from the system that stores them and the software that reports and analyses them.
- **NIST Special Publication 800-92:** This publication guides computer security log management.
- **Azure Monitor Logs:** Guidelines for log monitoring on Microsoft Azure.
- **Google Cloud Logging:** Guidelines for logging on the Google Cloud Platform (GCP).
- **Oracle Cloud Infrastructure Logging:** Guidelines for logging on the Oracle Cloud Infrastructure (OCI).
- **Virginia Tech - Standard for Information Technology Logging:** Sample log review and compliance guideline.

Log Collection

Log collection is an essential component of log analysis, involving the aggregation of logs from diverse sources such as servers, network devices, software, and databases.

For logs to effectively represent a chronological sequence of events, it's crucial to maintain the system's time accuracy during logging. Utilising the **Network Time Protocol (NTP)** is a method to achieve this synchronisation and ensure the integrity of the timeline stored in the logs.

As this is a foundational step to ensuring that a security analyst would have a comprehensive data set to review, the following is a simple step-by-step process to achieving this, bearing in mind the need to prioritise the collection based on significant information:

- **Identify Sources:** List all potential log sources, such as servers, databases, applications, and network devices.
- **Choose a Log Collector:** Opt for a suitable log collector tool or software that aligns with your infrastructure.
- **Configure Collection Parameters:** Ensure that time synchronisation is enabled through NTP to maintain accurate timelines, adjust settings to determine which events to log at what intervals, and prioritise based on importance.
- **Test Collection:** Once configured, run a test to ensure logs are appropriately collected from all sources.

Log Management

Efficient Log Management ensures that every gathered log is stored securely, organised systematically, and is ready for swift retrieval. A hybrid approach can provide a balanced solution by hoarding all log files and selectively trimming.

Once you've collated your logs, effective management of them is paramount. These steps can be followed to achieve this:

- **Storage:** Decide on a secure storage solution, considering factors like retention period and accessibility.
- **Organisation:** Classify logs based on their source, type, or other criteria for easier access later.
- **Backup:** Regularly back up your logs to prevent data loss.
- **Review:** Periodically review logs to ensure they are correctly stored and categorised.

Log Centralisation

Centralisation is pivotal for swift 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.

Centralising your logs can significantly streamline access and analysis. Here's a simple process for achieving it:

- **Choose a Centralised System:** Opt for a system that consolidates logs from all sources, such as the Elastic Stack or Splunk.

- **Integrate Sources:** Connect all your log sources to this centralised system.
- **Set Up Monitoring:** Utilise tools that provide real-time monitoring and alerts for specified events.
- **Integration with Incident Management:** Ensure that your centralised system can integrate seamlessly with any incident management tools or protocols you have in place.

Log Storage

Logs can be stored in various locations, such as the local system that generates them, a centralised repository, or cloud-based storage.

The choice of storage location typically depends on multiple factors:

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

Log Retention

It is vital to recognise that log storage is not infinite. Therefore, a reasonable balance between retaining logs for potential future needs and the storage cost is necessary. Understanding the concepts of Hot, Warm, and Cold storage can aid in this decision-making:

Managing the cost of storing logs is critical for organisations, and carefully selecting Hot, Warm, or Cold storage strategies can help keep these costs in check.

- **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.
- **Warm Storage:** Logs from **six months to 2 years**, acting as a data lake, easily accessible but not as immediate as Hot storage.

- **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

Log deletion must be performed carefully to avoid removing logs that could still be of value. The backup of log files, especially crucial ones, is necessary before deletion.

It is essential to have a well-defined deletion policy to ensure compliance with data protection laws and regulations. Log deletion helps to:

- Maintain a manageable size of logs for analysis.
- Comply with privacy regulations, such as GDPR, which require unnecessary data to be deleted.
- Keep storage costs in balance.

Best Practices: Log Storage, Retention and Deletion

- Determine the storage, retention, and deletion policy based on both business needs and legal requirements.
- Regularly review and update the guidelines per changing conditions and regulations.
- Automate the storage, retention, and deletion processes to ensure consistency and avoid human errors.
- Encrypt sensitive logs to protect data.
- Regular backups should be made, especially before deletion.

ogs are more than mere records of historical events; they can serve as a guiding compass. They are invaluable resources that, when skillfully leveraged, can enhance system diagnostics, cyber security, and regulatory compliance efforts. Their role in keeping a record of historical activity for a system or application is crucial.

Log Analysis Process

Log analysis involves Parsing, Normalisation, Sorting, Classification, Enrichment, Correlation, Visualisation, and Reporting. It can be done through various tools and techniques, ranging from complex systems like Splunk and ELK to ad-hoc methods ranging from default command-line tools to open-source tools.

Data Sources

Data Sources are the systems or applications configured to log system events or user activities. These are the origin of logs.

Parsing

Parsing is breaking down the log data into more manageable and understandable components. Since logs come in various formats depending on the source, it's essential to parse these logs to extract valuable information.

Normalisation

Normalisation is standardising parsed data. It involves bringing the various log data into a standard format, making comparing and analysing data from different sources easier. It is imperative in environments with multiple systems and applications, where each might generate logs in another format.

Sorting

Sorting is a vital aspect of log analysis, as it allows for efficient data retrieval and identification of patterns. Logs can be sorted by time, source, event type, severity, and any other parameter present in the data. Proper sorting is critical in identifying trends and anomalies that signal operational issues or security incidents.

Classification

Classification involves assigning categories to the logs based on their characteristics. By classifying log files, you can quickly filter and focus on those logs that matter most to your analysis. For instance, classification can be based on the severity level, event type, or source. Automated classification using machine learning can significantly enhance this process, helping to identify potential issues or threats that could be overlooked.

Enrichment

Log enrichment adds context to logs to make them more meaningful and easier to analyse. It could involve adding information like geographical data, user details, threat intelligence, or even data from other sources that can provide a complete picture of the event.

Enrichment makes logs more valuable, enabling analysts to make better decisions and more accurately respond to incidents. Like classification, log enrichment can be automated using machine learning, reducing the time and effort required for log analysis.

Correlation

Correlation involves linking related records and identifying connections between log entries. This process helps detect patterns and trends, making understanding complex relationships between various log events easier. Correlation is critical in determining security threats or system performance issues that might remain unnoticed.

Visualization

Visualization represents log data in graphical formats like charts, graphs, or heat maps. Visually presenting data makes recognising patterns, trends, and anomalies easier. Visualisation tools provide an intuitive way to interpret large volumes of log data, making complex information more accessible and understandable.

Reporting

Reporting summarizes log data into structured formats to provide insights, support decision-making, or meet compliance requirements. Effective reporting includes creating clear and concise log data summaries catering to stakeholders' needs, such as management, security teams, or auditors. Regular reports can be vital in monitoring system health, security posture, and operational efficiency.

Log Analysis Tools

Security Information and Event Management (SIEM) tools such as Splunk or Elastic Search can be used for complex log analysis tasks.

However, in scenarios where immediate data analysis is needed, such as during incident response, Linux-based systems can employ default tools like `cat`, `grep`, `sed`, `sort`, `uniq`, and `awk`, along with `sha256sum` for hashing log files. Windows-based systems can utilise EZ-Tools and the default cmdlet `Get-FileHash` for similar purposes. These tools enable rapid parsing and analysis, which suits these situations.

Additionally, proper acquisition should be observed by taking the log file's **hash during collection** to ensure its admissibility in a court of law.

Therefore, it is imperative not only to log events but also to ensure their integrity, that they are analysed, and any lessons obtained from the logs be learned, as the safety and efficiency of an organisation can depend on them.

Log Analysis Techniques

Log analysis techniques are methods or practices used to interpret and derive insights from log data. These techniques can range from simple to complex and are vital for identifying patterns, anomalies, and critical insights. Here are some common techniques:

Pattern Recognition: This involves identifying recurring sequences or trends in log data. It can detect regular system behaviour or identify unusual activities that may indicate a security threat.

Anomaly Detection: Anomaly detection focuses on identifying data points that deviate from the expected pattern. It is crucial to spot potential issues or malicious activities early on.

Correlation Analysis: Correlating different log entries helps understand the relationship between various events. It can reveal causation and dependencies between system components and is vital in root cause analysis.

Timeline Analysis: Analysing logs over time helps understand trends, seasonalities, and periodic behaviours. It can be essential for performance monitoring and forecasting system loads.

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.

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.

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.

These techniques can be applied individually or in combination, depending on the specific requirements and complexity of the log analysis task. Understanding and using these techniques can significantly enhance the effectiveness of log analysis, leading to more informed decisions and robust security measures.

Notes During the Room:

Important Takeaways:

Entry 6

Room Name: Wireshark Basics

Date Completed:
Notes During the Room:

Important Takeaways:

Entry 7

Room Name: Wireshark 101

Date Completed:
Notes During the Room:

Important Takeaways:

Entry 8

Room Name: Windows Fundamentals 1

Date Completed:

Notes During the Room:

Important Takeaways:

Entry 9

Room Name: Windows Fundamentals 2

Date Completed:
Notes During the Room:

Important Takeaways:

Entry 10

Room Name: Windows Fundamentals 3

Date Completed:
Notes During the Room:

Important Takeaways:

Entry 11

Room Name: Windows Forensics 1

Date Completed:

Notes During the Room:

Important Takeaways:

Entry 12

Room Name: Windows Forensics 2

Date Completed:
Notes During the Room:

Important Takeaways:

Entry 13

Room Name: Intro to Log Analysis

Date Completed:
Notes During the Room:

Important Takeaways:

Entry 14

Room Name: Splunk Basics

Date Completed:

Notes During the Room:

Important Takeaways:

Entry 15

Room Name: Incident Handling with Splunk

Date Completed:
Notes During the Room:

Important Takeaways:

Entry 16

Room Name: Splunk 2

Date Completed:
Notes During the Room:

Important Takeaways:

Entry 17

Room Name: Splunk 3

Date Completed:

Notes During the Room:

Important Takeaways: