TryHackMe Journal - Anthony Awoyele

Entry 1

Room Name: Linux Fundamentals 1

Date Completed:
Notes During the Room:
Task 1 - Introduction:

- Linux is a popular operating system used in various devices and systems worldwide.
- This room covers some Linux history and helps start the journey as a Linux user.
- Mentions essential commands to interact with the file system and understand users and groups in Linux.

Task 2 - A bit of background on Linux:

- Linux is more intimidating than Windows but has its advantages.
- Linux powers websites, car entertainment systems, PoS systems, and critical infrastructures.
- There are many distributions (flavors) of Linux, such as Ubuntu and Debian, which are open-source and highly extensible.
- Ubuntu Server can run on systems with only 512MB of RAM.

Task 3 - Interacting with your first Linux machine:

- This room provides an Ubuntu Linux machine that can be interacted with in the browser.
- To start the machine, click the "Start Machine" button on the top-right of the task.
- A card will appear with the machine's information, including the IP address and expiry timer.
- Remember to "Terminate" the machine once done with the room.

Important Takeaways:

Entry 2

Room Name: Linux Fundamentals 2

Date Completed:
Notes During the Room:
Objectives:

- Utilize SSH to manage terminal access remotely.
- Explore using flags and arguments with commands.
- Learn to copy, move, and manage file access.

Key Concepts:

1. SSH (Secure Shell):
    - Protocol for encrypted communication between devices.
    - Allows remote execution of commands.
    - Encryption ensures data privacy in transmission.

2. Command Usage Example:
    - ssh user@MACHINE_IP: Connect to a remote machine with specified username and IP.

3. Flags and Arguments:
    - Augment commands for extended functionality.
    - Example: ls shows directory contents; adding -a displays hidden files.

4. Manual Pages (Man Pages):
    - Access using man command for documentation.
    - Provides detailed options and usage examples for commands.

5. Commands for File System Interaction:

Command	Full Name	Purpose
touch	touch	Create a new file
mkdir	make dir	Create a new directory
cp	copy	Copy files or directories
mv	move	Move or rename files or directories
rm	remove	Remove files or directories
file	file	Identify file types

    - Example Usage:
        - touch file: Create a blank file named "file".
        - mkdir dir: Create a directory named "dir".
        - rm -R dir: Remove directory "dir" recursively.

6. Permissions and Access:
    - Access defined by Read, Write, Execute permissions.
    - Important commands like ls -l show detailed file permissions and access.

7. User and Group Management:
    - su user: Switch to a different user account.
    - su -l user: Login as a user with their home environment.

8. Notable Directories:

Directory	Purpose
/etc	Stores essential system configuration files, e.g., passwd
/var	Stores variable data like logs, databases
/root	Home directory for root user
/tmp	Temporary files, cleared on system reboot

9. Real-world Application:
    - Permissions allow granularity between different users and groups.
    - /tmp is writable by default, useful for temporary storage needs.

Important Takeaways:

1. SSH Protocol:
    - Enables secure, encrypted remote access and command execution on other machines.
    - Essential for managing systems over a network securely.

2. Command Line Proficiency:
    - Understanding flags and arguments is crucial for leveraging the full capabilities of commands.
    - Manual pages are valuable resources for learning command extensions and options.

3. File System Operations:
    - Basic operations like creating, moving, copying, and deleting files/folders are fundamental skills.
    - Commands such as touch, mkdir, cp, mv, and rm are frequently used in file management.

4. Permission and User Management:
    - File and directory permissions dictate access levels for users and groups.
    - Mastering user switching and understanding group permissions is critical for system security and multi-user environments.

5. Critical System Directories:
    - Directories like /etc, /var, /root, and /tmp each serve unique roles that are vital to system operation and administration.
    - Knowing the purpose and typical contents of these directories aids in effective system navigation and management.

Entry 3

Room Name: Linux Fundamentals 3

Date Completed:
Notes During the Room:
Objectives:

- Explore utilities for everyday use like text editors, file transfer, and package management.

- Develop skills in automation, process management, and system logging.

Key Concepts:

1. Terminal Text Editors:
    - Nano:
        - Simple, user-friendly text editor.
        - Use nano filename to create/edit files.
        - Supports basic functionality: search, copy/paste, navigate lines.
    - VIM:
        - Advanced, customizable text editor with syntax highlighting.
        - Modular and adaptable for development environments.

2. File Downloading with Wget:
    - Downloads files from the web via HTTP.
    - Usage: wget <URL>

3. File Transfers with SCP (Secure Copy):
    - Uses SSH to securely transfer files between systems.
    - Format: scp SOURCE DESTINATION

4. Serving Files with Python HTTPServer:
    - Lightweight, easy-to-use server to share files.
    - Usage: python3 -m http.server (runs on port 8000 by default).

5. Process Management:
    - Viewing Processes:
        - ps: List running processes.
        - ps aux: Show processes from all users and system processes.
        - top: Real-time process monitoring and statistics.
    - Managing Processes:
        - kill <PID>: Terminate a process with specific signals (e.g., SIGTERM, SIGKILL).
        - Background (&) or foreground (fg) processes for multitasking.

6. Automation and Scheduling with Crontabs:
    - Schedule tasks using formats involving MIN, HOUR, DOM, MON, DOW, CMD.
    - Example: 0 */12 * * * cp -R /home/user/Documents /var/backups/
    - Useful tools: Crontab generators and editors.

7. Package Management:
    - Use Ubuntu's apt system for installing/removing software.
    - Manage repositories with add-apt-repository.
    - Secure installation with GPG keys.

8. System Logging:
    - Logs located in /var/log directory for monitoring processes and activities.
    - Critical for performance diagnosis and security audits.

Command	Full Name	Purpose
nano	Nano	Text editing with a simple interface
vim	Vi Improved	Advanced text editing with customizable features
wget	Wget	Download files from the web via HTTP
scp	Secure Copy	Secure file transfer between systems using SSH
python3 -m http.server	Python HTTPServer	Serve files over HTTP from a local directory
ps	Process Status	Display information about currently running processes
top	Top Command	Real-time system process monitoring
kill	Kill	Terminate processes using process IDs
crontab	Cron Table	Schedule periodic tasks based on time intervals
apt	Advanced Package Tool	Manage software packages on Ubuntu systems

Important Takeaways:

1. Text Editors:
   - Nano is suitable for beginners, while VIM offers advanced capabilities.
   - Essential for managing and editing configuration files and scripts.

2. File Transfer and Serving:
   - Wget and SCP facilitate transfers, Python's HTTPServer shares files via network.
   - Practical for software download, file sharing, and web/API interaction.

3. Process Management:
   - Understand process IDs, usage of ps/top to assess system performance.
   - Use appropriate signals for efficient process control.

4. Automation with Crontabs:
   - Simplifies recurring tasks, enhancing efficiency and reliability.
   - Customize schedules to meet specific operational needs.

5. Package and Repository Management:
   - Apt enhances software management through easy installation/removal.
   - Understand repository handling for expanding OS capabilities.

6. System Logging:
   - Logs help in system troubleshooting, security monitoring, and tracking user activities.

Entry 4

Room Name: Linux Strength Training

Date Completed: July 29th, 2024

Notes During the Room:

Objectives:

1. Develop proficiency in finding files and directories based on specific criteria such as size, user, and modification date.

2. Understand and apply encryption, hashing, and encoding techniques for data security.

3. Manage and manipulate file operations efficiently within a Linux environment.

4. Gain foundational skills in interacting with SQL databases, including viewing and organizing data.

Key Concepts:

1. File Searching with Find:
   - Use the find command to locate files based on parameters such as file name, size, user, date modified, etc.
   - Syntax flexibility allows targeted searches to enhance file management and auditing tasks.

2. Basic File Operations:
   - Utilize commands like cp, mv, touch, and mkdir for copying, moving, renaming, and creating files and directories.
   - Employ efficient syntax to perform operations on multiple files or directories simultaneously.

3. Security Techniques:
   - Explore the importance of hashing with tools like MD5 and SHA-256 for integrity checking and data validation.
   - Apply gpg for encrypting and decrypting sensitive files, and understand the principles of symmetric encryption.
   - Use brute-force methods, with tools like John the Ripper, to illustrate hash cracking.

4. Data Encoding:
   - Understand base64 encoding and decoding for converting binary data into an ASCII format.
   - Differentiate between encoding and encryption in terms of data transformation and security.

5. Introduction to SQL Databases:
   - Learn to start and stop MySQL services and connect to databases locally or remotely.
   - Execute SQL commands to display databases, tables, and retrieve data.

File Operations

Command	Full Name	Purpose
find	Find	Search files/directories based on various criteria
cp	Copy	Copy files or directories
mv	Move	Move or rename files or directories
touch	Touch	Create new files
mkdir	Make Directory	Create new directories

nano	Nano Editor	Open and edit files
cat	Concatenate	Display content of files
scp	Secure Copy	Transfer files securely to a remote machine

Security & Hashing

Command	Full Name	Purpose
gpg	GNU Privacy Guard	Encrypt/decrypt files
john	John the Ripper	Crack hashes using wordlists
base64	Base64 Encoding/Decoding	Encode or decode data in base64 format
Hash-identifier	Hash Identifier	Identify hash types

Database Management

Command	Full Name	Purpose
mysql	MySQL Client	Connect to MySQL databases
SHOW DATABASES;	Show Databases	List all available databases
USE	Use Database	Select a specific database to work with

SHOW TABLES;	Show Tables	List all tables in the selected database
DESCRIBE	Describe Table	Show structure of a table
SELECT * FROM	Select All From Table	Display all data from a specified table

Important Takeaways:

1. Effective File Management:
    - Mastering commands and syntax for file and directory manipulation is essential for efficient system administration.
    - Understanding the find command significantly aids in searching and auditing file systems.

2. Security Best Practices:
    - Hashing and encryption are vital for protecting data and ensuring integrity within systems and applications.
    - Be aware of the vulnerabilities in weaker hashing algorithms like MD5 and SHA1, and prefer stronger algorithms like SHA-256.

3. Data Conversion Skills:
    - Recognizing the need and methods for encoding and decoding data is crucial in handling system files that rely on different formats.

4. Database Interaction:
    - Basic SQL command fluency is critical for accessing and organizing data within databases.
    - Understanding relational databases and table structures underpins effective data management and usage.

Entry 5

Room Name: Intro to Logs

Date Completed:
Notes During the Room:

Objectives:

1. Understand the role of logs as records of historical activities to identify and mitigate potential threats.

2. Gain insights into various types of logs, logging mechanisms, and collection methods across multiple platforms.

3. Acquire hands-on experience in detecting and defeating adversaries through log analysis.

4. Learn to interpret and analyze logs using tools and techniques for better security posture and incident response.

Key Concepts:

1. Importance of Logs:
   - Logs serve as vital historical records to detect, analyze, and respond to security incidents.
   - Different types of logs provide insights into system operations and security status.

2. Log Types and Formats:
   - Common log types include application, audit, security, system, and network logs.
   - Log formats can be semi-structured, structured, or unstructured, impacting how logs are parsed and analyzed.

3. Log Management:
   - Efficient log storage, centralization, and management enhance organization's ability to perform in-depth analysis and rapid incident response.
   - Techniques like log rotation and compression optimize log storage space.

4. Log Collection:
   - Involves aggregating logs from diverse sources using tools like rsyslog.
   - Ensures synchronization and integrity of logs for effective analysis.

5. Log Analysis:
   - Involves parsing, normalization, sorting, classification, enrichment, correlation, visualization, and reporting.
   - Employs both complex systems like SIEM (e.g., Splunk, Elastic Search) and command-line tools for quick, efficient analysis.

Important Commands:

Command	Full Name	Purpose
syslog	Syslog Protocol	Standard for message logging across systems
logrotate	Log Rotate	Automates the rotation, compression, and management of log files
rsyslog	Enhanced Syslog	Advanced logging system for Linux
ntpdate pool.ntp.org	Network Time Protocol Date	Sync system time with NTP server
cat, grep, sed, awk	Various Unix Tools	Used for parsing and processing log files
sort, uniq	Sort & Unique	Organize and deduplicate log entries

Important Takeaways:

1. Logs as Historical Records:
   - Essential for assessing system health, compliance, and security.
   - Enable detection of trends, anomalies, and security threats.

2. Comprehensive Log Management:
   - Centralizing and managing logs ensures efficient retrieval and analysis.
   - Log rotation and categorization techniques optimize storage.

3. Log Analysis Techniques:
   - Use systematic approaches to derive insights and support decision-making.
   - Tools and techniques enhance the process from simple text parsing to advanced machine learning.

4. Security Tool Integration:
   - Integrating logs with Security Information and Event Management (SIEM) platforms enhances security monitoring.
   - Logs offer critical context for tools like Endpoint Detection and Response (EDR) and Intrusion Detection and Prevention Systems (IDPS).

5. Hands-On Proficiency:
   - Practical skills in log collection and analysis bolster incident detection and response capabilities.

- Encouraged to further explore advanced log analysis scenarios such as forensic investigations and threat hunting using industry tools.

Entry 6

Room Name: Wireshark Basics

Date Completed: july 29th, 2024
Notes During the Room:

Important Takeaways:

Entry 7

Room Name: Wireshark 101

Date Completed:
Notes During the Room:
Objectives:

1. Familiarize with Wireshark installation, live packet capturing, and PCAP analysis.

2. Understand packet filtering techniques to streamline analysis.

3. Gain practical skills in analyzing common network protocols.

4. Learn the forensic analysis of a known exploit using PCAP files.

Key Concepts:

1. Wireshark Basics:
   - Learn to navigate interfaces, manage capture filters, and capture live traffic.
   - Understand how to load and analyze PCAP files with Wireshark.

2. Packet Capture Techniques:
   - Overview of collection methods including network taps, MAC floods, and ARP poisoning.
   - Criteria for efficient setup in capturing network traffic.

3. Filtering and OSI Layers:
    - Use of filtering operators (and, or, eq) for precise packet analysis.
    - Understanding packet layers in relation to the OSI model.

4. Protocol Analysis:
    - ARP: Identify request/reply packets and analyze traffic sources.
    - ICMP, TCP, DNS, HTTP/HTTPS: Understand and analyze protocol-specific traffic details.
    - Zerologon exploit: Analyze PCAP of a known Active Directory exploit.

5. Additional Wireshark Features:
    - Utilize Wireshark's built-in visual features for enhanced analysis, such as protocol hierarchy and endpoint tracking.

Important Commands:

Command/Feature	Purpose
Wireshark Filtering	Narrow down packet analysis with filters for IP, protocols, and ports.
ip.addr==10.0.0.1	Example of IP address filtering in Wireshark
tcp.port eq 80	Example of TCP port filtering
ARP Analysis Filter	Identifying ARP requests and replies using Opcode fields.
Protocol Analysis	Detailed inspection of packet layers (1-5) based on OSI model.

Important Takeaways:

1. Wireshark Proficiency:
    - Essential for network packet analysis, threat identification, and forensics.
    - Allows comprehensive inspection of network traffic across various protocols.

2. Efficient Traffic Collection:
    - Proper setup and method selection ensure effective traffic monitoring.
    - Physical and logical collection strategies should be carefully considered and applied.

3. Filter Utilization:

- Filters enhance focus during analysis, enabling pinpointing of packets of interest.
- Differentiate between capture filters and display filters for effective usage.

4. Protocol Analysis Understanding:
   - Recognizing protocol-specific behaviors enhances the ability to identify anomalies.
   - ARP, ICMP, TCP, DNS, and HTTP(S) are foundational for network traffic comprehension.

5. Threat Intelligence Application:
   - Know common exploit signatures and network behaviors to detect compromises.
   - Apply threat intelligence to identify Indicators of Compromise (IOCs) in PCAP analysis.

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:
Objectives:

1. Understand the fundamentals of computer forensics, specifically on Windows systems.

2. Learn about the Windows Registry and its significance in forensic analysis.

3. Gain proficiency in using forensic tools to acquire and analyze registry data.

4. Identify key forensic artifacts related to system and user activity on Windows.

Key Concepts:

1. Forensic Artifacts:
    - Learn the significance of forensic artifacts which provide evidence of user activity.
    - Windows systems track user activities stored in different locations like the registry, user profiles, and application-specific files.

2. Windows Registry:
    - The Windows Registry is a crucial data source containing configuration information about hardware, software, and user data.

- It consists of keys and values organized into root keys and hives, which can provide a trail of user activity.

3. Registry Structure:
    - Understand the five root keys: HKEY_CURRENT_USER, HKEY_USERS, HKEY_LOCAL_MACHINE, HKEY_CLASSES_ROOT, HKEY_CURRENT_CONFIG.
    - Recognize important hives: DEFAULT, SAM, SECURITY, SOFTWARE, SYSTEM, NTUSER.DAT, and USRCLASS.DAT.

4. Data Acquisition Tools:
    - Learn about tools like KAPE, Autopsy, FTK Imager for data acquisition.
    - Understand how to acquire restricted files for analysis.

5. Registry Analysis Tools:
    - Tools like Registry Viewer, Zimmerman's Registry Explorer, and RegRipper are essential for analyzing registry data.
    - Focus on Zimmerman's tools for their capability to incorporate transaction logs for a complete registry view.

Important Commands/Tools:

Command/Tool	Purpose
regedit.exe	Access the Windows Registry editor
KAPE	Acquire registry data from live systems or disk images
Autopsy	Forensics platform for data acquisition and analysis
FTK Imager	Extract files from disk images or live systems
Registry Viewer	View and analyze registry hives
Registry Explorer	Comprehensive utility for registry inspection and analysis
RegRipper	Extracts key data from registry hives into a report

AppCompatCacheParser	Parses ShimCache data for application compatibility tracking

Important Takeaways:

1. Role of Forensic Artifacts:
   - Artifacts serve as evidence for reconstructing user actions and system changes.
   - Allows forensic investigators to build a timeline and determine the extent of activity.

2. Windows Registry as a Data Source:
   - Central repository for configuration and user preference data.
   - Key source for investigating system settings, installed applications, and user accounts.

3. Effective Data Acquisition:
   - Utilize reliable and forensically sound methods to extract data for analysis.
   - Tools like KAPE and FTK Imager facilitate data retrieval from live systems and disk images.

4. Analytical Tool Proficiency:
   - Master tools like Registry Explorer for thorough registry analysis.
   - Incorporate transaction logs for the most accurate and up-to-date data assessment.

5. Comprehensive Forensic Examination:
   - Identify system version, account information, network configurations, and connected devices.
   - Understand the forensic significance of registry paths and entries including recent files, user activity, and autorun programs.

Entry 12

Room Name: Windows Forensics 2

Date Completed:

Notes During the Room:

Objectives:

1. Understand different file systems, including FAT, exFAT, and NTFS, and their structures.

2. Learn about methods and tools for data recovery, particularly on Windows systems.

3. Explore locations and tools for extracting execution artifacts from file systems.

4. Gain insights into how to identify and analyze file and application usage on Windows.

Key Concepts:

1. File Systems Overview:
   - FAT (File Allocation Table): Basic file system with versions FAT12, FAT16, and FAT32.
   - exFAT: Developed for large files and removable storage, resolving FAT limitations.
   - NTFS (New Technology File System): Advanced file system with features like journaling, access controls, and alternate data streams.

2. NTFS Data Structures:
   - Master File Table (MFT) and its critical components ($MFT, $LOGFILE, $UsnJrnl).

3. Data Recovery Techniques:
   - Understanding concepts of disk images and recovering deleted files using Autopsy.
   - Disk images provide a forensic copy of storage devices, essential for non-invasive analysis.

4. Execution Artifacts:
   - Prefetch Files: Track last run times and execution details for applications.
   - Windows 10 Timeline: Captures recently executed applications.
   - Jump Lists & Shortcut Files: Track recently accessed applications and files.

5. Forensic Tools:
   - Employ tools like AFTECmd, PECmd, WxTCmd, and LECmd for parsing and analyzing artifacts.

Important Commands/Tools:

Command/Tool	Purpose
Autopsy	Recover deleted files and analyze disk images
MFTECmd	Parse MFT files for detailed volume analysis
PECmd	Parse Prefetch files for execution details

WxTCmd	Parse Windows 10 Timeline for application usage
JLECmd	Parse Jump Lists to find recent file usage
LECmd	Parse Shortcut files to determine first and last access details

Important Takeaways:

1. Understanding File Systems:
   - Knowledge of FAT, exFAT, and NTFS helps identify storage limitations and capabilities.
   - NTFS is superior for security, access control, and recoverability.

2. Effective Data Recovery:
   - Use forensic tools like Autopsy to non-invasively recover files and analyze disk images.
   - Disk imaging is crucial for preserving the integrity of original evidence.

3. Artifact Awareness:
   - Forensic analysis relies heavily on identifying and interpreting execution artifacts.
   - Artifacts like Prefetch files, Timeline data, and Jump Lists provide insight into user and application activity.

4. Forensic Tool Utilization:
   - Proficient use of tools like PECmd, MFTECmd, and WxTCmd enhances analysis of file and application usage.
   - Parsing tools assist in converting raw data into actionable insights on system behavior.

5. Execution and Access Tracking:
   - Shortcut files, Jump Lists, and browser histories are valuable for reconstructing user activity.
   - Combining information from multiple artifact sources paints a comprehensive picture in investigations.

Entry 13

Room Name: Intro to Log Analysis

Date Completed:
Notes During the Room:
Objectives:

1. Understand the importance of log analysis in cybersecurity and system monitoring.

2. Learn best practices and methodologies for effective log analysis.

3. Gain hands-on experience with essential tools for log analysis and threat detection.

4. Identify common patterns and anomalies in log data to detect security threats.

Key Concepts:

1. Log Analysis Importance:
   - Logs are crucial in detecting system errors, security incidents, and threat patterns.
   - Provides insights into system operations, user activities, and network interactions.

2. Log Types:
   - Various logs include application, audit, security, server, system, network, database, and web server logs.
   - Each log type provides unique insights into specific components of the infrastructure.

3. Analysis Methodologies:
   - Creating timelines and super timelines for incident response.
   - Data visualization and monitoring with tools like Splunk and Kibana.

4. Common Attack Patterns:
   - Recognize patterns of abnormal user behavior, SQL injection, cross-site scripting, and path traversal.
   - Automated and manual analysis techniques to identify threats.

5. Log Analysis Tools:
   - Use Sigma, YARA, and CyberChef for associative and automated event detection.
   - Utilize command-line tools like grep, awk, and sed for log parsing and filtering.

Important Commands:

Command	Full Name	Purpose
cat	Concatenate	Display contents of files
less	Less	View files page by page
tail	Tail	View the last part of a file

wc	Word Count	Count lines, words, and characters in a file
cut	Cut	Extract specific columns from files
sort	Sort	Arrange file data
uniq	Unique	Remove duplicate lines from input
sed	Stream Editor	Manipulate and transform text in files
awk	Awk	Pattern scanning and processing language
grep	Global Regular Expression Print	Search for specific patterns in files

Important Takeaways:

1. Logs as Investigative Tools:
    - Serve as detailed records of events and facilitate incident detection, response, and compliance reporting.
    - Enable threat hunting and system troubleshooting through structured analysis.

2. Tool-Assisted Analysis:
    - Use tools like CyberChef, Grok, and regex for advanced log parsing and analysis.
    - Employ dedicated analysis platforms like ELK Stack and Splunk for comprehensive data visualization and real-time monitoring.

3. Pattern Recognition:
    - Automate and streamline threat detection with Sigma and YARA rules.
    - Recognize and respond to common attack signatures and anomalies to manage security incidents efficiently.

4. Command-Line Proficiency:
    - Master in-depth log manipulation and analysis using Unix-based command-line utilities for swift responses.
    - Regular expressions enhance precision in extracting and processing log data.

5. Best Practices:
    - Regularly update methodologies and tools to adapt to evolving security landscapes.

- Embrace a layered approach using both automated and manual techniques for robust log analysis and threat detection.

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: