## LESSON: Mastering the Linux Command-Line Interface

### Primer

* Review the “Lesson Opener” and “Real World Scenario” with the learners prior to starting the module.
* Throughout the module, you will find “Consider the Real World Scenario” slides. Review the questions found on these slides, tie the concepts back to the scenario discussed at the start of the lesson as well as content you are presenting, and encourage the learners to share their thoughts.
* For each lesson, you will find a “Pulse Check” slide which is the opportunity for instructors to open a poll to gather feedback from the learners. Leave the poll open for about 1 minute and after you close the poll, share the results with the learners. Encourage the learners to share their thoughts. This information will help the instructors as well as the learners better understand where they are with regards to the lesson.
* Labs are to be demonstrated live for each module. The demonstration of labs is the top priority for the lead instructor. While demonstrating each lab, encourage students to participate and explore.
* At the end of each lesson, it is important to take a few minutes to review the key concepts for the lesson, provide guidance on what the learners can do to prepare for the next lesson, and wrap up with Q&A.

### Summary

In this lesson, learners will explore the Linux operators, mastering the art of combining commands for optimized executions and efficient workflows. They will become adept at using key commands like "Cat," "Head," and "Tail" to manipulate and filter data directly from the terminal. Understanding the distinct roles of STDIN, STDOUT, and STDERR data streams will be emphasized. Learners will gain proficiency in the versatile "echo" command for text display and file writing, utilizing redirection symbols effectively. Piping commands using | will become second nature, enhancing functionality. The powerful "grep" command's filtering prowess will be harnessed, often in conjunction with pipes. Learners will learn to locate files with precision using the find and locate commands, and explore advanced data manipulation through grep's flags and "awk" for extraction and printing. They will gain real time insights into system performance using the “top” command, and discover how to interact with it using keyboard shortcuts. The “ps” command's ability to inspect processes, including detached ones, will be covered, alongside the utility of the “history” command for recalling and executing past commands using the ! symbol.

### Objectives

* Identify advanced Linux operators.
* Explain how to combine commands.
* Illustrate the use of output filters.
* Explain how to use the Less and More commands to read files within the terminal.
* Categorize data streams in Linux.
* Describe the use of the echo command to write and read files.
* Define the concept of piping.
* Illustrate the use of the grep and awk commands for output filtering.
* Explain the use of the locate and find commands for file searching.
* Explain the functioning and key features of the top command.
* Illustrate the use of the ps command.
* Explain the functioning of the history command.

### Lesson Activities and Teaching Strategies

|  |  |  |
| --- | --- | --- |
| Estimated Time | Lesson Portion | Directions |
| 2 | **Lesson Opener:**  Mastering Linux CLI | * Introduce learners to the importance of Mastering Linux CLI in cybersecurity * Highlight the benefits of using Linux Advanced commands and provide examples |
| 2 min | **Lesson Companion:** Overview | * Provide an overview about the lesson companion and highlight the key points that will be discussed in the class |
| 5 | **Real World Scenario:**  Scaling With Linux | * Review the real world scenario challenge and inform learners that you will be constantly coming back to this scenario throughout the lesson to discover how to solve and apply concepts to this real situation. |
| 20 min | **Cyber Uncovered:**  Linux Advanced Operators and Output Filtering | * Explain the role of Linux operators in customizing and optimizing command execution. * Emphasize that these operators allow the combination and execution of commands based on specific conditions. * Introduce key operators: ; (semicolon), || (Or), && (And), and & (Background). * Describe the purpose and behavior of each operator in relation to command execution. * Highlight how these operators affect the sequence and conditions of command execution. * Provide examples of using the semicolon operator to execute multiple commands simultaneously, highlighting their independence. * Illustrate the behavior of the Or operator in executing the following command only if the previous one fails. * Demonstrate the And operator's functionality, executing commands based on the success of the previous command. * Explain the Background operator and its use in running independent commands concurrently. * Discuss how operators enable the execution of command combinations. * Explain the concept of executing commands conditionally based on the success of the previous command. * Show how these operations can be used in conjunction with bash conditionals to enhance their effectiveness. * Walk through examples of using the Background operator to open a browser in the background. * Show the process of sequential execution using the semicolon operator to display user, hostname, and working directory. * Demonstrate conditional execution with the And && operator, creating a directory and moving to it if the creation was successful. * Introduce the concept of managing output using commands like cat, head, and tail. * Explain how these commands enhance the ability to view and manipulate text data. * Clarify the primary purposes of the cat, head, and tail commands. * Describe the advanced filtering technique using the -n flag with head and tail commands. * Provide examples of using the -n flag to specify the number of lines to display. * Illustrate the precise control over data viewing achieved through this technique. * Explain the purpose of the more command as a basic text pager. * Detail the key commands used to navigate through content while using the more command. * Introduce the less command, highlighting its enhanced navigation features. * Walk through the usage of the more command over a specific file (e.g., /etc/passwd). * Demonstrate how to move between pages and lines while using the more command. * Illustrate the extended capabilities of the less command, including forward and backward navigation. * Be prepared to discuss the implication of the real world scenario presented at the beginning of class on “Leveraging Linux for Streamlined Operations**”**. There are specific prompts that you should ask learners to reflect on to apply this concept to the real world scenario. |
| **5 min Break** | | |
| 5 min | **Real World Scenario:**  Mastering Linux CLI | * Review the real world scenario challenge, ask the learners to share their perspective * Share your own perspective and provide examples * Inform the learning that you will be constantly coming back to this scenario throughout the lesson to discover how to solve and apply concepts to this real situation. |
| 20 min | **Cyber Uncovered:**  Input, Output, and Redirection | * Start by explaining the concept of data streams in Linux. * Describe the three main categories: STDIN, STDOUT, and STDERR. * Emphasize the role of each stream in command execution and system communication. * Define STDIN as standard input, involving user inputs and command arguments. * Explain STDOUT as standard output, comprising user-readable outputs and command results. * Describe STDERR as a special output for errors and exceptions in the terminal. * Demonstrate an example of STDIN, STDOUT, and STDERR and provide a real life applications for each category * Introduce the "echo" command's common use for text display. * Detail how the echo command can write to files using > and >> redirection. * Show examples of overwriting and appending text to files using these redirection symbols. * Introduce piping, represented by the "|" symbol, as a powerful concept. * Explain how piping links output from one command as input to another and recap on STDIN, STDOUT and how it is used in conjunction with piping. * Provide real world examples of using piping to chain commands. * Demonstrate combining "cat" with "more" for efficient content viewing. * Run the command "cat file.txt | more" to show a step-by-step output. * Illustrate how "more" displays content page by page, enhancing readability. * Introduce the "grep" command as a tool for filtering command output. * Explain its role as a line filter for displaying specific content. * Highlight common use cases of "grep" in combination with pipes. * Compare the "find" and "locate" commands for file searching. * Detail "find" as real time search with precise filtering options. * Explain "locate" as a faster option using a pre-built database, requiring occasional updates. * Introduce advanced usage of "grep" with additional flags. * Describe flags like -i (case-insensitive), -n (line number), and -r (recursive). * Encourage learners to access the manual page for more flag information using "man grep." * Define AWK as a data extraction tool in Linux systems. * Describe its role in extracting specific data from text. * Explain how AWK can be used to print specific columns from a file using $X parameter * Walk through an example of using AWK to filter and print file permissions and filenames. * Illustrate the combination of ls, grep, and AWK using pipes to process file information. * Show the practicality of this approach for file analysis without needing advanced knowledge. * Highlight that there are other tools as well such as cut command * **Give learners a 5 minute break (coming up in the next section)** * Be prepared to discuss the implication of the real world scenario presented at the beginning of class on advanced linux commands “part of this migration strategy includes training the employees to work with advanced Linux commands, further leveraging the capabilities of the system and aligning with the company's technological progression” (lesson companion). |
| **5 min Break** | | |
| **10 min Pulse Check** | | |
| 20 min | **Cyber Uncovered:**  System Monitoring Commands | * Begin by explaining the importance of monitoring a Linux system's performance. * Introduce the top command as a powerful real time monitoring tool. * Highlight its ability to display active processes, CPU usage, memory utilization, and other vital statistics. * Describe the information displayed by the top command. * Detail the breakdown of processes, CPU usage, memory usage, and system uptime. * Emphasize the default sorting of processes by CPU usage. * Explain that learners can interact with the top command using keyboard shortcuts. * Mention specific shortcuts like 'P' to sort processes by CPU usage and 'M' to sort by memory usage. * Show examples of how these shortcuts modify the displayed information. * Present a sample output of the top command. * Discuss the various sections displayed, including processes, CPU usage, memory utilization, and uptime. * Interpret the data in the output and help learners understand the significance of each section. * Introduce the ps command as a tool for displaying running processes. * Explain its basic functionality of showing processes and their process IDs (PIDs). * Provide examples of using the ps command without options to display processes in the current terminal session. * Introduce the ps -aux command for a more comprehensive view of processes. * Describe the purpose of options like -a, -u, and -x in the command. * Provide a visual example of using ps -aux to display detailed information about all running processes. * Explain the significance of the history command in Linux. * Describe how it lists previously executed commands from the command-line history. * Emphasize its usefulness for recalling and re-executing recent commands. * Detail the usage of the "!" symbol followed by a command number from the history to quickly re-execute specific commands without retyping. * Be prepared to discuss the implication of the real world scenario presented at the beginning of class regarding improvement of resource monitoring with respect to CloudTech. There are specific prompts that you should ask learners to reflect on to apply this concept to the real world scenario. |
| 20 min | **Lab:**  Obtain System Information | * Remind learners to use this lab to practice and apply the concepts they have learned throughout the day. * Learners will receive direct feedback on their lab to properly assess their knowledge and determine where they might need additional assistance. * Perform a lab demonstration and contextualize the tasks in the lab with the topics you covered * Explain certain command in the lab that are not covered during the lecture such as block devices and explain what they are and the purpose of knowing block devices in Linux |
| 20 min | **Lab:**  Linux Pipelines | * Remind learners to use this lab to practice and apply the concepts they have learned throughout the day. * Learners will receive direct feedback on their lab to properly assess their knowledge and determine where they might need additional assistance. * Task 7- 10 covers different use cases for awk command. Review these sections thoroughly and provide learners different real world applications for these commands |
| 10 min | **Lesson Closure** | * Encourage learners to read ahead of time * Provide learners additional resources to read / practice and assign homework (e.g., future labs) before you demonstrate the labs during the next class * Spend some time to highlight what are the key takeaways from today’s lesson * Important topics covered during the class includes   + Combining commands using Linux operators helps to efficiently handle complex tasks and automation   + Linux output filters such as head and tail   + Highlight use cases for less and more   + Standard input/output and standard error   + Use cases for less-than and greater-than symbols in Linux   + Piping, grep, and awk commands for data manipulation   + System monitoring commands such as top, ps, and htop   + History command and the usage of the ! symbol followed by a specific command number from the history command output * Q&A |
|  | **Additional Time Filler (if needed)** | * Review using Kahoot or other similar platforms * Conduct interview preparation conversations * Continue discussions on real-world scenarios * Demonstrate how to create users in Linux and grant them permissions * Discuss different career paths in cybersecurity and highlight the roles that require Linux skills |