List of Experiments

Academic Year: 2025-26
Program Name: M.Sc. Digital Forensics and Information Security
Semester: I
Paper Name and Code: Python and Scripting
(CTMSDFIS SI L3)
No. of Credits: 2 (Lab)

1. On Syllabus record: Shell Scripting

- Aim: To automate the process of local user account creation using Bash scripting.

- Materials Used: Linux system, users.csv file, terminal, bash shell.

- Procedure: Write a bash script to read usernames from a CSV file and create system accounts if they don't already exist. Generate and display a summary log of created and skipped users in a report file.

- Result: Students will learn how to automate user creation using shell scripts, implement conditional logic, read and write files, and generate logs using bash scripting in a Linux environment.

2. On Syllabus record: Shell Scripting

- Aim: To analyze authentication logs for suspicious login attempts using a bash script.

- Materials Used: Linux system, /var/log/auth.log, terminal, bash shell, mail utility (optional).

- Procedure: Write a script to parse system logs, identify repeated failed login attempts, and log suspect IPs. Save them to a file and optionally send alerts if thresholds are exceeded.

- Result: Students will learn to automate security log parsing, detect intrusion patterns, and respond using scripting techniques.

3. On Syllabus record: Advanced Shell Scripting

- Aim: To efficiently search for files using regex patterns, archive results, and automate backup using shell scripting.

- Materials Used: Linux system, terminal, bash shell, find, grep, tar, gzip utilities.

- Procedure: Write a shell script to search directories for files matching regular expression criteria, archive the results, and compress the archive. Validate directory existence, handle errors if files are not found, and log each operation for troubleshooting.

- Result: Students gain experience with practical use of file search, regular expressions, automation of archiving/backup, error handling, and scripting best practices.

4. On Syllabus record: Advanced Shell Scripting

- Aim: To automate network information gathering and present results in a well-formatted, printable report.

- Materials Used: Linux system, terminal, bash shell, ifconfig/ip, ping, grep, awk, lp command.

- Procedure: Develop a shell script to collect network interface details, check connectivity to key servers, and format the collected data into a human-readable report. Include script sections using branching (if), loops (for/while), arrays for server lists, and allow user input for specific diagnostics.

- Result: Students will be able to automate system checks, gather and process networking data, format output for printing, use arrays, and implement structured shell scripts with flow control and user interaction.

5. On Syllabus record: Python Fundamentals

- Aim: To understand Python variables, data structures, conditional statements, and function creation.

- Materials Used: Python 3.x installed, text editor (e.g., VS Code, Sublime), terminal or IDE (IDLE, Thonny, or PyCharm).

- Procedure: Write a Python program that takes input from the user, stores it in variables, and performs basic operations using lists, dictionaries, and conditional statements. Create functions to display records and use selection (if-else) and iteration (for, while) to process and filter the data.

- Result: Students learn to write Python programs using variables, lists, dictionaries, conditionals, iteration, and functions with reusable logic.

6. On Syllabus record: Python Fundamentals

- Aim: To implement object-oriented programming concepts in Python and perform basic client-server communication using sockets.

- Materials Used: Python 3.x installed, terminal/IDE, two Python programs (client and server), internet connection for testing.

- Procedure: Write Python classes to represent network devices using attributes and methods, then apply inheritance to create specialized device types. Develop simple client-server socket programs that send and receive messages, and demonstrate class usage during data exchange.

- Result: Students will understand OOP principles (classes, objects, inheritance), apply them in a networking context, and write socket-based client-server applications in Python.

7. On Syllabus record: Python Forensics

- Aim: To understand file I/O, handle exceptions effectively, and perform basic disk image and image forensics using Python.

- Materials Used: Computer with Python 3.x, terminal or IDE, sample files (text and image), os module, Python Imaging Library (PIL/Pillow).

- Procedure: Write a Python script to safely read data from disk and image files, implementing exception handling for missing/corrupt files. Use the PIL module to extract pixel data from an image, analyze basic statistics (e.g., max or mean pixel value), and log results to a file.

- Result: Students gain practical experience in safe file access, exception handling, and leveraging libraries for disk/image forensics, building foundational forensic automation skills in Python.

8. On Syllabus record: Python Forensics

- Aim: To analyze captured network packets, perform geolocation acquisition, and extract blacklists from network evidence using Python.

- Materials Used: Computer with Python 3.x, sample PCAP (packet capture) file, scapy library, GeoIP or third-party geolocation database, text editor or IDE.

- Procedure: Write a Python script that loads a PCAP file, extracts IP addresses from packets, and determines their geolocation. Add logic to flag and log any addresses on a predefined blacklist, and output flagged activity for further analysis.

- Result: Students will learn how to automate network packet parsing, perform geolocation of endpoints, and apply blacklist/whitelist logic, enabling efficient forensic triage of network evidence.

9. On Syllabus record: PowerShell Scripting

- Aim: To demonstrate the use of PowerShell remoting to invoke remote commands and process output using objects and advanced functions.

- Materials Used: Two Windows machines (or local system with remoting enabled), PowerShell ISE, administrative privileges.

- Procedure: Write a script using Invoke-Command to run diagnostics remotely on another system and parse the output into objects. Create advanced functions using param, leverage -ByRef arguments, and handle output using Select-Object, hash tables, and error record objects.

- Result: Students will understand PowerShell remoting, output processing, passing values by reference, and scripting best practices for remote system administration.

10. On Syllabus record: PowerShell Scripting

- Aim: To automate reading and writing to text and CSV files in PowerShell with proper error handling.

- Materials Used: Windows PC, PowerShell (v5.1+), PowerShell ISE or Visual Studio Code.

- Procedure: Write a PowerShell script that reads a list of users from a CSV file, processes the data, and writes results to a new file with error handling using $?, $Error, and try/catch. Explore use of param blocks, special variables, loops, arrays, and file I/O with graceful error capture and logging.

- Result: Students will gain practical experience automating CSV/text file operations, using PowerShell special variables and implementing structured error handling in scripts.

Approved By:

Course Instructor **Program Coordinator** **Dean Academic**