CSET213 – Linux and Shell Programming
Course Type - Core L-T-P Format 2-0-4 Credits – 4

COURSE SUMMARY

This course describes the essential ideas behind the open-source operating system approach to programming. Knowledge of Linux and shell script helps to understand the backbone of cybersecurity. This course involves basic Linux commands, Shell scripting, File structure and management, Processes, Inter-process communication, Socket programming, and security.

COURSE-SPECIFIC LEARNING OUTCOMES (CO)

CO1: To articulate Linux commands that are used to manipulate system operations at an admin level.
CO2: To write Shell Programming using Linux commands.
CO3: To design and write applications to manipulate internal kernel-level Linux File systems.

Detailed Syllabus

Module 1 (Contact hours: 8)
Linux and Linux utilities, Architecture of Linux, features of Linux, Introduction to vi editor, Linux commands, File handling utilities, security by file permissions, process utilities, disk utilities, Networking commands, Text processing utilities, backup utilities, User management.

Module 2 (Contact hours: 8) ✓ bash
Shells need and types, Derived Operators, Linux session, Standard streams, Redirection, Pipes, Tee command, command execution, command-line editing, Quotes, command substitution, job control, aliases, variables, shell/environment customization, Filters, and pipes, File operations.

Module 3 (Contact hours: 12) ✓
Grep Operation, Grep Commands, Grep Address, Grep Application, Sed Scripts, operation, Unix file structure, File descriptors, System calls and device drivers, File management, File structures, System calls for file management, Directory API, Process and Process Structure, Process table, Viewing processes, System processes, Process scheduling, zombie processes, orphan process, Fork and its operation, Signals functions, unreliable signals, interrupted system calls, Signal sets, File locking, Threats and Vulnerabilities analysis of Linux- direct, indirect, veiled, conditional, Security Measures in Linux-SSH key pair, Scan Log files , Close Hidden ports, Linux Malwares- Botnets, Ransomware, Rootkits, Socket, Socket communications, UDP, TCP, AWK, Shell Scripting and Security- Password Tester, Permissions and Access Control Lists, Shell Scripting for DevOps- Using environment variables, Bash Script

STUDIO WORK / LABORATORY EXPERIMENTS:

Students will use LINUX / UBUNTU to gain hands-on experience on LINUX and Shell programming, Linux commands, their uses and practice, editors: vi, nano etc, Introduction to Shell, Shell basic commands, variables Shell programming environments- filters and pipe, Shell programming File handling, Grep its use and commands. Using of Grep with pipe and filters, Unix file structuring, inodes and related system calls. File handling commands and API, Network Penetration testing tools, Wireshark, Nmap, Hash cat, Process management, creation, termination and other useful commands, Process scheduling. Parent, zombie and orphan process, Process system calls. Fork, exec, wait and signal, various commands. Basics of Socket Programming via UDP socket

TEXTBOOKS/LEARNING RESOURCES:

1) M. Ebrahim and A Mallett, Mastering Linux Shell Scripting: A Practical Guide to Linux Command-Line, Bash Scripting, and She (2 ed.), Packt Publication, 2018. ISBN 978-1788990554.
2) R. Blum and C. Bresnahan, Linux Command Line and Shell Scripting Bible (3 ed.), Wiley, 2016. ISBN 978-1118983843.

REFERENCE BOOKS/LEARNING RESOURCES:

1) W.R. Stevens, UNIX Network Programming (3 ed.), PHI Publications, 2017. ISBN 978-8120307490.

Lecture wise Plan

Lect. No.	Content Planned	Content Practiced
1.	Course structure/handout Assessment mechanism (15) Linux and Linux Utilities (35)	Done As given
2.	Architecture of Linux (20) features of Linux (15) VI editor (15)	Done As given
3.	File handling utilities (15) security by file permissions (10) process utilities (15)	Filesystem, Linux Security
4.	disk utilities (20) Networking commands (25)	As given
5.	Text processing utilities (25) backup utilities (20)	As given
6.	User Management (45)	+ group management
7.	Linux session (15) Standard streams (15) Redirection (10) Pipes (10)	done
8.	**Assessment /Buffer Lecture***	
9.	Shell/environment customization (30) Filters and pipes (15)	done
10.	File operations (45)	done
11.	Grep Operation (20) Grep Commands (30)	done
12.	Case study (50)	
13.	Start-ups in Linux and Shell programming (50)	
14.	Unix file structure (45)	done
15.	File descriptors (20) System calls and device drivers (30)	done
16.	**Assessment/ Buffer Lecture**	
17.	File management (15) File structures (30)	done
18.	Process and Process Structure (10) Process table (20) Viewing processes (10) System processes (10)	done
19.	Process scheduling (40) zombie processes, orphan process (10)	done
20.	Fork and its operation (45)	done
21.	Signals functions (20) unreliable signals (15) interrupted system calls (15)	Done
22.	Signal sets (25) File locking (25)	Done
23.	Threats and Vulnerabilities analysis of Linux (10) Direct (10) Indirect (10) Veiled (10) Conditional (10)	Done

24.	Security Measures in Linux (5) SSH key pair (15) Scan Log files (15) Close Hidden ports (15)	
25.	Linux Malwares (10) Botnets (15) Ransomware (10) Rootkits (10)	
26.	Trojans, Viruses, Worms (10) Shell Scripting and Security (5) Password Tester (15) Permissions and Access Control Lists (20)	
27.	Shell Scripting for DevOps (15) Using environment variables (15) Bash Script (20)	
28.	**End Term Assessment/Buffer**	

Lab wise Plan

Lab No.	Planned Lab Content	Practiced
1.	Installation of Linux and its various distros	Lab Assignment 1: Installation of Ubuntu as dual boot
2.	Linux commands- commands and editor	Lab Assignment 2
3.	Linux Commands-User Management	Lab Assignment 3
4.	Linux commands-Networking Commands	Lab Assignment 4
5.	Introduction to Shell, Shell basic commands, variables	Lab Assignment 5
6.	Shell Programming operators	Lab Assignment 6
7.	Shell programming loops	Lab Assignment 7
8.	Shell programming environments- filters and pipe	Lab Assignment 8
9.	Shell programming File handling	Lab Assignment 9
10.	Shell Functions	Lab Assignment 10
11.	AWK commands and its application	Lab Assignment 11
12.	Grep its use and commands. Using of Grep with pipe and filters	Lab Assignment 12
13.	Unix file structuring, inodes and related system calls.	Lab Assignment 13
14.	**Mid-Term Lab Assessment/Buffer**	
15.	**Mid-Term Lab Assessment/Buffer**	
16.	File handling commands and API	Lab Assignment 14
17.	Network Penetration testing tools, Wireshark, Nmap, Hash cat	Lab Assignment 15
18.	Threats and Vulnerabilities analysis of Linux	Lab Assignment 19
19.	Process management, creation, termination and other useful commands, Process scheduling. Parent, zombie, and orphan process	Lab Assignment 16
20.	Process system calls. Fork, exec	Lab Assignment 17

21.	Wait and signal	Lab Assignment 18
22.	Inter Process communication via pipes and shared memory, various commands.	Lab Assignment 20
23.	Socket Programming via UDP socket	Lab Assignment 21
24.	~~Shell Scripting and Security~~	~~Lab Assignment 22~~
25.	~~Shell Scripting for DevOps~~	~~Lab Assignment 23~~
26.	**Buffer Lab**	
27.	**End Term Lab Assessment/Buffer**	
28.	**End Term Lab Assessment /Buffer**	

Assessment Component

S. No	Component	Marks
1.	Continuous Lab Assessment	20
2.	Linux Kernel-based Project/Hackathon	20
3.	Certification	10
4.	End Term	35
5.	Mid Term	15

MOOCs

S. No	Course Name	Platform
1.	Linux - Learn App Development using Linux	Infosys Springboard: https://rb.gy/gqzifd
2.	Kali Linux - The Ultimate Kali Linux and Penetration Testing Training	Infosys Springboard: https://rb.gy/gqzifd
3.	Operating Systems and You: Becoming a Power User	Coursera https://www.coursera.org/learn/os-power-user
4.	Open Source Software Development, Linux and Git Specialization	Coursera

To be Filled each Semester

1) Probable Industry Talks:

S. No	Talk Title	Resorce Person	Company
1	Talk-1 (After 2nd Module)	TBA	TBA
2	Talk-2 (After 3rd Module)	TBA	TBA

2) Relevant MOOC Courses being Referred: Yes
3) Probable Case Studies: Yes
4) Advanced Research Topics: No
5) Start-ups to be discussed: Yes
6) Assessment Components Details: Yes
7) Software required : Ubuntu OS, Kali, Docker
8) Hardware required: A machine with Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz OR similar, 8GB RAM, 500GB HDD
9) Industry/certificate mapping: Yes