GWENDAL LE COGUIC

COURSES

Each training session focuses on practical demonstrations, interactive exercises, and discussions based on real-life situations to ensure the acquisition of key skills.

Each session can be taken:

- on-site or remotely
- in French or English
- individually or in groups

Quotes available upon request

Tel: +33 7 69 02 25 10
Email: contact@glc.st

1. HACKING & WEB APPLICATION SECURITY 4
a. Description 4
b. Educational goals 4
c. Prerequisites 4
d. Targeted audience 4
e. Program 5
Day 1 - The basics of web security 5
Day 2 - User input flaws 6
Day 3 - Authenticated user vulnerabilities 6
2. ADVANCED HACKING & WEB APPLICATION SECURITY 8
a. Description 8
b. Educational goals 8
c. Prerequisites 8
d. Targeted audience 8
e. Program 9
Day 1 9
Day 2 9
3. CYBERSECURITY AWARENESS 10
a. Description 10
b. Educational goals 10
c. Prerequisites 10
d. Targeted audience 10
e. Program 11
4. LINUX HARDENING 12
a. Description 12
b. Educational goals 12
c. Prerequisites 12
d. Targeted audience 12
e. Program 13
5. INTRODUCTION TO PENTESTING 14
a. Description 14
b. Educational goals 14
c. Prerequisites 14
d. Targeted audience 15
e. Program 15
Day 1 - Introduction and information gathering 15
Day 2 - Exploitation techniques and web security 15
Day 3 - Post-exploitation and password attacks 16
6. SÉCURITÉ LARAVEL 17
a. Description 17
b. Educational goals 17
c. Prerequisites 17
d. Targeted audience 17
e. Program 17

7. BUG BOUNTY OVERVIEW....20
a. Description....20
b. Educational goals....20
c. Prerequisites....20
d. Targeted audience....20
e. Program....21
8. PHP FOR BEGINNERS....22
a. Description....22
b. Educational goals....22
c. Prerequisites....22
d. Targeted audience....22
e. Program....23
Day 1 - PHP basics....23
Day 2 - OOP and database....24
Day 3 - External libraries....24
9. THE BASICS OF SQL....25
a. Description....25
b. Educational goals....25
c. Prerequisites....25
d. Targeted audience....25
e. Program....26
Day 1 - SQL fundamentals....26
Day 2 - Advanced SQL concepts....26
10. INTRODUCTION TO BASH SCRIPTING....27
a. Description....27
b. Educational goals....27
c. Prerequisites....27
d. Targeted audience....27
e. Program....28
Day 1 - Introduction to Bash and file manipulation....28
Day 2 - Control structures, functions, and advanced scripting....28
Day 3 - Process management, automation, and security....29

1. HACKING & WEB APPLICATION SECURITY



a. Description

*This intensive 3-day training is designed for **developers** and **system administrators** aiming to enhance the security of their web applications. Combining **theoretical knowledge** with **practical exercises**, the course covers **common vulnerabilities**, utilization of **essential tools**, **best practices** in secure web development. Participants will learn to **identify**, **exploit**, and **remediate** security flaws, gaining hands-on experience to proactively defend against real-world threats.*

b. Educational goals

- Understand the main web application vulnerabilities
- Learn how to detect and exploit common security flaws
- Use tools like *Burp Suite* and *SQLmap* effectively
- Apply secure coding practices to prevent attacks
- Improve overall application resilience against threats

c. Prerequisites

- Basic knowledge of web development (HTML, CSS, JavaScript)
- Familiarity with web protocols (HTTP, HTTPS)
- Experience with using command-line tools

d. Targeted audience

- Web developers
- System administrators
- Dev(Sec)Ops engineers
- Any cybersecurity enthusiasts

e. Program

Day 1 - The basics of web security

The landscape of web insecurity

- Web security: a vast topic
- Architecture of a typical Web app
- Open Web Application Security Project (OWASP)
- CWEs, CVEs & CVSS
- External and internal attackers
- The different steps of an attack
- An attacker's bag of tricks
- Common misconceptions

Security's place in the development cycle

- The importance of inventory
- Shadow IT
- Production bias & incompatibility with Agile approaches
- The idea of *DevSecOps*
- The importance of testing & code reviews
- Static & dynamic analysis

Outside help

- Security audits and *Bug Bounty*
- Vulnerability disclosure policy

HTTP protocol

- Reminder about the HTTP(s) protocol
- Anatomy of HTTP requests & responses
- Semantics & status codes
- Introduction to *Dev Tools*
- Introduction to *Burp Suite*

URLs

- Anatomy of a URL
- Open redirects exploitation

Disclosure of sensitive information

- Sequential IDs
- Verbose error messages
- Directory listing
- Clear-text logging of sensitive information
- Secrets on the client side
- API Excessive data exposure
- Public *Git* repositories & history
- *Git* directory deployed

Day 2 - User input flaws

Input validation

- Client-side, server-side, or both?
- HTTP parameter pollution
- Mass assignment

Cross-Site Scripting (XSS)

- Cross-site scripting is everywhere
- Types of cross-site scripting
- What is the impact?
- How to protect & how to bypass

SQL Injection (SQLi)

- What is it?
- What is the impact?
- How to protect & how to bypass
- Presentation of *SQLMap*
- *NoSQL* injection

Content Security Policy (CSP)

- Definition
- Most common directives
- Possible configuration
- What is the impact?
- How to protect & how to bypass

Other forms of injection

- HTML injection
- Code injection
- OS command injection

Files

- Common file issues
- Arbitrary file upload
- Path traversal
- Local & remote file inclusion

Day 3 - Authenticated user vulnerabilities

Authentication & vulnerabilities

- Authentication
- User enumeration
- Rate limiting and security questions
- Multiple-factor authentication

Passwords

- Cryptography in a nutshell: encoding, hashing, encryption
- Password-strength policy
- Leaked passwords
- Password storage
- Cracking tools

Cookies

- Cookies attributes: *secure, httponly, domain, samesite…*
- Session hijacking

Cross-Site Request Forgery (CSRF)

- What is it?
- What is the impact?
- How to protect & how to bypass

Clickjacking

- What is it?
- What is the impact?
- How to protect & how to bypass

Authorization & access control

- Authentication vs. authorization
- Roles and permissions
- Insecure direct object reference

Cross-Origin Resource Sharing (CORS)

- Refresher on the *Same-Origin Policy*
- Possible misconfigurations
- What is the impact?
- How to protect

2. ADVANCED HACKING & WEB APPLICATION SECURITY



a. Description

*An intensive 2-day training designed for professionals with a solid foundation in web security. It delves into advanced vulnerabilities and equips participants with the skills to **identify**, **exploit**, and **mitigate complex security issues** in web applications.*

b. Educational goals

- Identify and exploit advanced web vulnerabilities
- Understand complex attack vectors like SSRF, XXE, and prototype pollution
- Evaluate risks linked to third-party components and infrastructure
- Apply secure coding practices to mitigate real-world threats

c. Prerequisites

- Basic knowledge of web development (HTML, CSS, JavaScript)
- Familiarity with web protocols (HTTP, HTTPS)
- Experience with using command-line tools
- Familiar with basic web vulnerabilities (XSS, CSRF, SQLi…)

d. Targeted audience

- Web developers
- System administrators
- Dev(Sec)Ops engineers
- Any cybersecurity enthusiasts

e. Program

Day 1

- Carriage Return Line Feed injection (CRLF)
- Second order bugs
- Prototype pollution
- *PostMessage*
- Server-Side Request Forgery (SSRF)

Day 2

- Dependency confusion
- Subdomain takeover
- JSON Web Token security (JWT)
- XML External Entity injection (XXE)
- Emails security

3. CYBERSECURITY AWARENESS



a. Description

*This one-day awareness course is designed for all **employees and collaborators** who interact with digital tools in their daily work. The goal is to demystify cybersecurity, present the **major threats and risks** facing organizations today, and empower participants with concrete, practical knowledge to reduce vulnerabilities. Through clear **explanations**, **real-world examples**, and **actionable advice**, this session helps create a culture of shared responsibility around information security.*

b. Educational goals

- Understand the main cybersecurity threats and their impact
- Identify common attack techniques
- Recognize the importance of user behavior in preventing incidents
- Apply basic best practices in daily work
- React appropriately to potential security incidents
- Understand legal and organizational responsibilities

c. Prerequisites

None.

d. Targeted audience

- Employees at all levels: managers and team leaders, human resources, administrative staff…
- IT Support and technical teams
- Remote Workers or telecommuters

e. Program

Introduction to cybersecurity

- What is cybersecurity and why it matters
- Real-world examples of threats
- Importance of user behavior

Organizational roles & responsibilities

- Who does what in cybersecurity
- Shared responsibilities and interdepartmental cooperation
- Risk awareness and internal communication

Legal and regulatory context

- Overview of GDPR and other key regulations
- What the law says about data protection and incident response
- Individual obligations and corporate compliance

Threat landscape & attack scenarios

- Common cyber threats
- Case studies and real incidents
- Understanding the human factor in cyberattacks

Best practices for secure behavior

- Password hygiene and multi-factor authentication
- Email security and identifying phishing attempts
- Safe browsing and social media usage

Securing digital tools & environments

- Safe use of devices (computers, phones, USBs)
- Working remotely securely
- Data backup and software updates

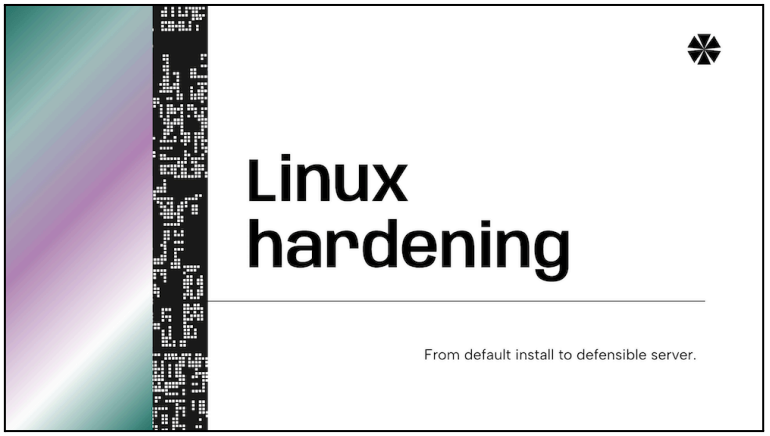
Responding to a cyber incident

- How to detect and report suspicious activity
- Immediate steps to take during a security breach
- Building a personal action plan

Wrap-up & recommendations

- Summary of key takeaways
- Resources for staying informed
- Open discussion and Q&A

4. LINUX HARDENING



a. Description

This intensive 1-day training teaches you how to harden a Linux server from the ground up. You'll learn to **minimize the system footprint**, **lock down user privileges**, **secure network exposure**, **and reinforce critical services**. Through hands-on exercises, you'll apply **real-world hardening** techniques used by security professionals.

b. Educational goals

- Identify common entry points, privilege-escalation paths, and misconfigurations
- Minimize installed software and apply kernel and filesystem protections
- Configure strong authentication, secure sudo policies, and controlled user environments
- Secure network interfaces, firewall rules, and remote access pathways
- Deploy logging, auditing, and basic intrusion-detection mechanisms

c. Prerequisites

- Being comfortable with CLI mode
- Familiarity with basic Linux administration

d. Targeted audience

- System administrators
- Dev(Sec)Ops engineers

e. Program

Introduction

- Why hardening matters?
- Understanding the attack surface
- Defense in depth
- Principle of least privilege
- Minimal install philosophy
- Hardening is role-specific
- Reference frameworks
- Baseline vs. hardening vs. compliance

System baseline

- Partitioning strategy
- Unnecessary packages & services
- Unused kernel modules
- Hardening the Kernel

Account hardening

- Mandatory Access Control (MAC)
- Account and sudo hardening
- Password and authentication policy

Network hardening

- Network exposure
- Firewalling
- Brute-force mitigation

Services hardening

- Dedicated users
- File permissions
- Service sandboxing

Filesystem, permissions & integrity

- SUID/SGID auditing
- Sticky bit and ACLs
- Immutable attributes
- File integrity monitoring

Logging, auditing & monitoring

- Kernel-level security event
- Centralized logging
- Intrusion detection concepts

Automation & compliance validation

- Scoring with Lynis
- Formal compliance scoring with OpenSCAP
- Hardening-as-code with Ansible

Resources

- Books
- Interactive platforms
- Courses & videos
- Tools

5. INTRODUCTION TO PENTESTING



a. Description

*The Pentest training is an intensive 3-day program designed to introduce **developers**, **system administrators** and any **cybersecurity enthusiasts** to the fundamentals of **ethical hacking**. Through a combination of **theory** and **practical exercises** in dedicated environments, participants will learn how to **identify and exploit vulnerabilities** in web applications and network services. Essential tools such as **Kali Linux**, **Metasploit**, **Burp Suite** and more will be covered. At the end of the training, participants will be able to conduct effective security audits and strengthen the protection of their IT infrastructures.*

b. Educational goals

- Understand the fundamentals of ethical hacking and legal boundaries
- Identify and analyze common vulnerabilities in web and network systems
- Use industry-standard tools like *Kali Linux, Metasploit,* and *Burp Suite*
- Perform manual exploitation techniques
- Conduct vulnerability assessments and interpret scan results
- Apply pentesting methodologies to simulate real-world attacks

c. Prerequisites

- Basic knowledge of IT systems and networks
- Familiarity with operating systems (Linux and Windows)
- Basic understanding of programming or scripting is a plus but not required

d. Targeted audience

- Developers
- System administrators
- Dev(Sec)Ops engineers
- Any cybersecurity enthusiasts

e. Program

Day 1 - Introduction and information gathering

Audit charter

- What is it?
- Content of the audit charter

Methodology

- Steps of an attack
- An attacker's bag of tricks
- Testing process

Introduction to Kali Linux

- What is it?
- Tools included
- Lab configuration

Information gathering

- Passive vs. active recon
- *Google, Shodan, GitHub* hacking
- Collect Companies and employees data
- Technologies
- ASN enumeration
- DNS enumeration, resolution and probing
- Port scan

Vulnerability scan

- CWEs, CVEs & CVSS
- Scanners

Day 2 - Exploitation techniques and web security

Report

- What is it?
- Content of the report

HTTP testing

- Requirements
- What would you test?
- Browser extensions
- Introduction to *Burp Suite*
- Cross-Site Scripting
- SQL injection
- Arbitrary file upload

Other network services

- Shell vs. reverse shell
- Find, install and use exploits
- Introduction to *Metasploit*
- Service exploitation: *FTP, SMB, PostgreSQL, WebDAV…*

Day 3 - Post-exploitation and password attacks

Final presentation

- What is it?

Password attacks

- Cryptography refresher: encoding, hashing, encryption
- Leaked passwords and default passwords
- Online services & network attacks
- Wordlists
- Linux vs. Windows

Privilege escalation

- Horizontal vs. Vertical
- Linux vs. Windows

Post-exploitation

- Persistency
- Antivirus evasion
- Port forwarding and port tunneling

6. SÉCURITÉ LARAVEL



a. Description

*This one-day course is designed for **Laravel developers** who want to build secure applications from the ground up. You'll learn to **identify common vulnerabilities**, **apply Laravel's built-in protections**, and **adopt best practices** to secure authentication, **authorization**, **input validation**, and more.*

b. Educational goals

- Recognize common web vulnerabilities and Laravel's protections
- Implement secure authentication and authorization with Laravel tools
- Configure CORS and CSRF protections properly
- Validate and sanitize user input securely
- Handle file uploads and storage with security in mind

c. Prerequisites

- Good experience with PHP and Laravel framework
- Understanding of web application security concepts
- Familiarity with basic web vulnerabilities (XSS, CSRF, SQLi…)

d. Targeted audience

- Laravel developers
- Security professionals
- Dev(Sec)Ops engineers

e. Program

Global configuration

- Basic PHP security

- Global configuration
- Sessions
- *APP_KEY*

The basics
- Raw PHP in views
- Inputs validation
- Logging and Error handling
- Updates and Backups
- Tricking bots

Cookies & sessions
- Global configuration
- Laravel specific
- Session fixation

Passwords & encryption
- Encrypt/decrypt data
- Passwords validation
- Reset passwords

Authentication & permissions
- Official packages
- Laravel passport
- Gates and policies
- Users / roles / permissions

Rate limiting
- Usage
- Ratelimit actions / routes

File uploads & file storage
- What is it?
- Validation rules and Sanitization
- Storages configuration and usage
- Authorization / permissions / visibility

Mass assignment
- What is it?
- Expectation vs. reality
- How to protect?

Excessive data exposure
- What is it?
- How to protect?
- Take away

SQL Injection (SQLi)

- What is it?
- Prepared statements
- Parameter binding
- Dangers
- Take away

Cross-Site Scripting (XSS)

- What is it?
- Controllers
- Templates
- Input sanitization
- External helps

Cross-Site Request Forgery (CSRF)

- What is it?
- CSRF token
- Middleware
- Forms
- *X-csrf-token* and *X-xsrf-token*

Cross-Origin Resources Sharing (CORS)

- What is it?
- Configuration
- *allowed_origins* and *supports_credentials*

7. BUG BOUNTY OVERVIEW



a. Description

This half-a-day course offers a complete introduction to the world of bug bounty. ***Tailored for directors****,* ***CISO****,* ***team leader****, this session covers everything from the history and structure of bug bounty programs to real-world vulnerability reporting and triaging. Through* ***case studies****,* ***best practices****, and* ***industry insights****, participants will leave equipped with a strong foundation to begin (or improve) their bug bounty journey,* ***both from the hacker and company perspectives****.*

b. Educational goals

- Understand what a *bug bounty* program is
- Learn how *bug bounty* fits into a security strategy
- Identify key roles and responsibilities
- Know how to handle vulnerability reports
- Evaluate your organization's *bug bounty* readiness

c. Prerequisites

- Basic understanding of cybersecurity principles
- Familiarity with vulnerability management and risk assessment

d. Targeted audience

- Cybersecurity managers and security teams
- CISO and risk management professionals
- Organizations new to *bug bounty*

e. Program

Welcome & icebreaker

- Instructor intro & participant backgrounds
- What to expect from the course
- What is *bug bounty* and why it matters?

Understanding *bug bounty*: the ecosystem

- History and evolution of programs
- Roles and motivations: companies, platforms, hackers, 3rd parties
- *Pentest* vs. *bug bounty*: key differences

Anatomy of a *bug bounty* program

- How programs are structured
- Public vs. private programs
- What companies expect vs. what hackers expect

Reports: triaging & managing conflicts

- Lifecycle of a report
- Handling duplicates and disputes

Policy & program management

- What companies should prepare before launching
- Program evolution: scope, rewards, privacy
- Common mistakes to avoid

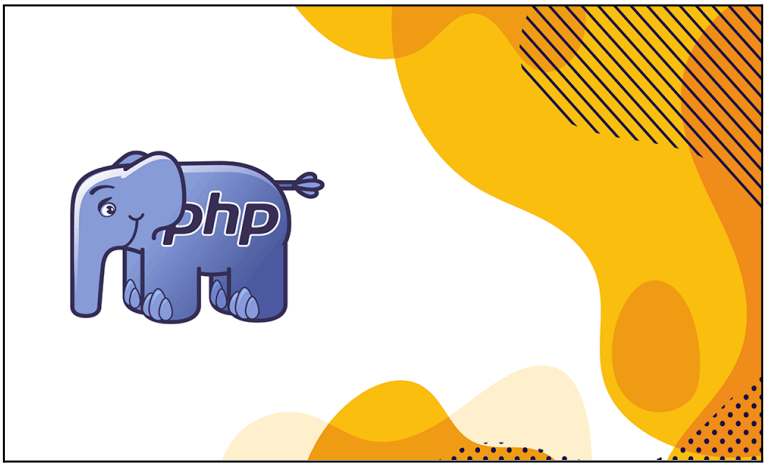
Rules, ethics & real case studies

- The unofficial rules of *bug bounty*
- Safe harbor, disclosure policies
- Failures and successes: *Verizon, Shopify, Zomato…*

Wrap-up & Q&A

- Recap of the day
- Open Q&A and personalized advice

8. PHP FOR BEGINNERS



a. Description

*A comprehensive 3-day training designed for beginners aiming to **build dynamic web applications using PHP**. Participants will learn to write structured PHP code, handle data with SQL, and integrate external libraries using Composer. The course emphasizes **practical learning through progressive exercises**, culminating in the development of a functional web application. By the end, attendees will be equipped to design and develop the server-side of a website, generating the necessary code for the front-end. This training is **ideal for novice developers** seeking to create applications or services based on the PHP programming language.*

b. Educational goals

- Understand PHP syntax and basic programming concepts
- Build dynamic web pages using PHP and HTML
- Manage data with MySQL and SQL queries
- Handle forms and user input securely
- Structure code using functions
- Use Composer to manage PHP libraries and dependencies

c. Prerequisites

- Basic knowledge of HTML and CSS
- Familiarity with web browsers and text editors

d. Targeted audience

- Beginners in web development
- Aspiring backend developers

- Anyone interested in PHP/MySQL to build dynamic websites

e. Program

Day 1 - PHP basics

What is PHP?

- A bit of History
- What is PHP and what is a PHP File?

PHP basics

- PHP tags & short tags
- PHP + HTML
- Comments

Variables

- Variables types
- Constants
- Quotes
- Concatenation
- Operators

Control Structures

- *if, elseif, else, while, for, while, for, foreach*
- *break, continue*
- *switch…case*
- *match*
- *include, require*

Functions

- Recursion
- Types declaration
- Variable arguments
- Default parameters
- Anonymous functions
- Arrow functions

HTTP methods

- Superglobals
- *GET* and *POST*
- Other HTTP methods
- Forms and filters

Persistency and storage

- Sessions
- Cookies
- File storage

Day 2 - OOP and database

The essence of Object Oriented Programming

- Basics
- Properties
- *Constructor* and *destructor*
- Keyword *$this*
- Visibility
- Constants
- Inheritance
- *Abstract*, *Final* and *Static*
- Magic methods
- Errors and exceptions
- HTTP headers
- Interfaces and traits
- Namespaces

Database

- Mysqli
- PDO

Day 3 - External libraries

External libraries

- Autoload
- Composer
- Dotenv
- PHPUnit
- PHP CLI

Ecosystem

- Extensions
- Frameworks
- CMS

9. THE BASICS OF SQL



a. Description

*This hands-on training is designed to give developers, analysts, and IT professionals a solid foundation in SQL. You'll learn how to **design databases**, **write efficient queries**, and **analyze data** using real-world exercises and best practices. Whether you're just getting started or looking to deepen your skills, this course will make you fluent in relational database querying.*

b. Educational goals

- Understand the structure and logic of relational databases
- Design and create normalized database schemas
- Write basic and advanced SQL queries to retrieve and manipulate data
- Perform data aggregation and analysis using SQL functions
- Optimize queries and apply best practices for performance

c. Prerequisites

- Basic understanding of data and its organization (spreadsheets, tables)
- Familiarity with fundamental programming or logic concepts (variables, conditions)

d. Targeted audience

- Developers
- Data analysts
- System administrators

e. Program

Day 1 - SQL fundamentals

Introduction to SQL and relational databases

- Theoretical explanation of the relational model
- Database architecture and components
- Basics of SQL: syntax, structure, and execution

Database design and schema creation

- Understanding ER diagrams and normalization
- Creating tables: *CREATE TABLE*, constraints (*PRIMARY KEY*, *FOREIGN KEY*, etc.)
- Data types and indexing

Basic SQL queries

- Retrieving data with *SELECT*, filtering with *WHERE*
- Using operators (*AND, OR, LIKE, IN*)
- Sorting results with *ORDER BY*

Day 2 - Advanced SQL concepts

Advanced queries and joins

- Understanding *INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN*
- Combining results with *UNION* and *INTERSECT*

Aggregations and subqueries

- Grouping data with *GROUP BY*, filtering with *HAVING*
- Using aggregate functions (*COUNT, SUM, AVG*, etc.)
- Writing subqueries with *EXISTS, NOT EXISTS*

Window functions

- Introduction to window functions (*OVER, PARTITION BY*)

Performance optimization

- Optimizing queries with indexing
- Common pitfalls and best practices

10. INTRODUCTION TO BASH SCRIPTING



a. Description

*This 3-day hands-on course introduces participants to Bash scripting for task automation and system administration. From writing **simple scripts** to **managing processes**, **securing code**, and **working over SSH**, you'll learn practical skills to boost productivity and control over your Linux environment. Perfect for **devs** and **sysadmins**.*

b. Educational goals

- Understand the fundamentals of the Bash shell and scripting syntax
- Write and execute basic to intermediate Bash scripts
- Automate common system tasks using loops, conditions, and variables
- Manage processes and schedule tasks
- Handle user input, signals, and logging in scripts

c. Prerequisites

- Being comfortable with CLI mode
- Familiarity with basic Linux commands

d. Targeted audience

- System administrators
- Dev(Sec)Ops engineers
- Anyone interested in automation

e. Program

Day 1 - Introduction to Bash and file manipulation

Introduction to Bash

- What is Bash?
- Why learn Bash?
- Basic commands

Writing and Running Bash scripts

- What is a Bash script?
- Create and execute a script
- The *Shebang*
- Grant permissions
- Run a script

Variables

- Understanding variables
- Declaring and using variables
- Types of variables

Conditional statements

- What are conditional statements?
- *if-else* statements
- *case* statements

Loops (*for, while, until*)

- *for* loops
- *while* loops
- *until* loops

Functions in Bash

- Why use functions?
- Defining and calling functions

Day 2 - Control structures, functions, and advanced scripting

Handling user input and arguments

- Reading user input
- Script arguments

Redirection and pipelines

- Output and error redirection
- Using pipelines

File and directory operations

- Creating and manipulating files/directories

Permissions and access rights

- Understanding Linux file permissions
- View and change permissions

Debugging and error handling

- Debugging
- Handling errors
- Handle script interruptions

Day 3 - Process management, automation, and security

Process management in Bash

- Understanding processes in Linux
- Viewing processes

Scheduling tasks

- Automating tasks
- *cron* job format

Logging and monitoring scripts

- Logging
- Checking system logs

Handling signals and traps

- What are signals?
- How to handle signals

Securing Bash scripts

- File permissions and execution restrictions
- Validating user input
- Using *sudo* securely

Working with remote servers and SSH

- Connect to a remote server
- Copying files securely
- Automating SSH tasks