"Python Cybersecurity Essentials: Entry-Level Security Specialist Training"

Course Introduction:

The "PCES – Certified Entry-Level Security Specialist with Python" course is designed to equip beginners with foundational skills in cybersecurity using Python. Over the span of seven intensive days, participants will gain hands-on experience in identifying and mitigating security vulnerabilities, understanding security protocols, and automating security tasks with Python. By the end of this course, learners will have a solid grasp of entry-level security concepts and practical Python programming skills tailored to cybersecurity applications.

Day 1: Introduction to Cybersecurity and Python Basics

• Overview of Cybersecurity Landscape: Explore the current trends, threats, and opportunities in the world of cybersecurity.

• Understanding Cybersecurity Terminology: Familiarize with essential terms and concepts used in the cybersecurity domain.

• Setting Up Python Environment: Learn to install Python and set up a development environment for security scripting.

• Basic Python Programming: Introduction to Python syntax, variables, data types, and basic input/output operations.

• Writing First Python Program: Develop a simple Python application to reinforce basic programming concepts.

Day 2: Networking Fundamentals and Python Networking

• Networking Basics: Understand the fundamentals of networking, including IP addresses, ports, and protocols.

• Python for Networking: Utilize Python libraries such as socket and requests to perform basic network operations.

• Introduction to Network Scanning: Learn about network scanning techniques and their role in cybersecurity.

• Building a Network Scanner with Python: Create a simple network scanner to identify active devices on a network.

• Ethical Considerations in Network Scanning: Discuss the legal and ethical aspects of network scanning activities.

Day 3: Understanding Security Threats and Vulnerabilities

• Common Security Threats: Explore various types of security threats including malware, phishing, and DDoS attacks.

• Vulnerability Assessment Basics: Learn about vulnerability assessment and its importance in securing systems.

• Python for Vulnerability Scanning: Use Python to identify potential vulnerabilities in systems and applications.

• Hands-on: Implementing a Basic Vulnerability Scanner: Develop a Python script to detect common vulnerabilities.

• Case Study: Analyzing Real-World Security Breaches: Examine recent security breaches to understand vulnerabilities and preventive measures.

Day 4: Introduction to Cryptography

• Cryptography Fundamentals: Understand the principles and importance of cryptography in security.

• Symmetric and Asymmetric Encryption: Learn about different encryption techniques and their use cases.

• Implementing Encryption with Python: Use libraries like PyCrypto to encrypt and decrypt data.

• Practical Exercise: Securing Data with Python: Write Python scripts to securely encrypt and decrypt files.

• Exploring Hash Functions: Understand how hash functions work and implement hashing in Python to ensure data integrity.

Day 5: Web Application Security and Python

• Overview of Web Application Security: Identify common web application vulnerabilities such as SQL injection and XSS.

• Using Python for Web Security Testing: Leverage Python tools like BeautifulSoup and Selenium for security testing.

• Developing a Simple Web Scraper: Use Python to scrape web data ethically and understand web security implications.

• Hands-on: Detecting SQL Injection Vulnerabilities: Write scripts to test web applications for SQL injection vulnerabilities.

• Introduction to Web Application Firewalls (WAFs): Understand the role of WAFs in protecting web applications.

Day 6: Automating Security Tasks with Python

• Benefits of Automation in Cybersecurity: Explore how automation enhances efficiency in security operations.

• Scripting Security Tasks with Python: Write Python scripts to automate repetitive security tasks.

• Building a Simple Intrusion Detection System: Develop a Python-based IDS to monitor network traffic and detect anomalies.

• Automating Report Generation: Create scripts to automatically generate security reports from scan results.

• Case Study: Successful Automation in Cybersecurity: Analyze how companies have effectively utilized automation to improve security posture.

Day 7: Practical Applications and Course Wrap-Up

• Applying Python in Real-World Security Scenarios: Discuss various scenarios where Python can be applied in cybersecurity.

• Final Project: Developing a Security Tool: Integrate learned concepts to develop a comprehensive security tool using Python.

• Best Practices in Python Security Scripting: Review best practices for writing secure and efficient Python scripts.

• Career Pathways in Cybersecurity: Explore potential career paths and further learning opportunities in cybersecurity.

• Course Review and Next Steps: Reflect on key learnings and discuss next steps for certification and professional development.