Mastering Python for Cybersecurity

Duration: *4 Day(s)*

Course Overview

Ideal for experienced security professionals venturing into Python, this hands-on course takes you from scripting fundamentals to sophisticated file manipulations and Python's extensive module capabilities. Specifically tailored for security tasks like log analysis and forensics, attendees will enhance their Python skills for use in real-world security scenarios, including application audits and defensive coding practices.

Review this course online at https://www.alta3.com/courses/A3-PS4894

Objectives

- Develop Python scripts adhering to best practices.
- Utilize Python's built-in data types effectively.
- Analyze logs and enhance security audits.
- Securely manage files and networks using Python.

Who Should Attend

- Security Analysts
- IT Professionals
- System Auditors
- Cybersecurity Enthusiasts

Prerequisites

- Basic programming skills
- Familiarity with Unix/Linux, Mac, or Windows

Course Outline

Introduction to Python

1. Overview of Python's history and benefits
2. How to find help with pydoc

Setting Up Python

3. Launching Python and utilizing the interpreter
4. Running scripts on different OS platforms
5. Overview of editors and IDEs

Beginning with Basics

6. Variable usage and conversion
7. Exploring built-in functions

8. String and number operations

Control Structures

9. Understanding flow control and whitespace
10. Applying conditional expressions and loops

Handling Sequences

11. Lists, tuples, and slicing techniques
12. Iterating through sequences and using comprehensions

File Operations

13. Working with text and binary files
14. Structuring binary data handling

Dictionaries and Sets

15. Creating and manipulating dictionaries and sets
16. Iteration techniques

Function Definitions

17. Constructing functions and understanding scope
18. Implementing nested functions

Sorting Techniques

19. Using sorted() and lambda functions
20. Sorting complex collections

Errors and Exceptions

21. Syntax and handling exceptions
22. Employing try-catch blocks

Modules and Packages

23. Importing modules and using packages
24. Understanding search paths and aliases

Object-Oriented Programming

25. Defining classes and constructors
26. Exploring methods and properties

Regular Expressions

27. RE syntax and object usage
28. Matching, searching, and replacing

Leveraging the Standard Library

29. Utilizing system and string modules
30. Handling CSV data

Date and Time Management

31. Parsing and handling timestamps

File System Operations

32. Managing paths, directories, and file attributes
33. Implementing security-based file access

Network Services

34. Retrieving web content securely

Developing Secure Applications

35. Safe data handling and static analysis
36. Incorporating command-line options

Log File Examination

37. Manipulating raw log files
38. Customizing security tools with Python

Crafting Security Filters

39. Detecting SQL-Injection attacks
40. Operating ModSecurity CRS

Conducting Packet Analysis

41. Sniffing packets with Python

Advanced Analytics

42. Executing security logging and analytics

Additional Content

43. Implementing RESTful services with Flask

44. Enhancing Python application security