

Python Security | Introduction to Python Programming for Security Analysts & Professionals

Implement Better Defenses in Python | Python Essentials, File Operations, Binary Data, Networking Services, Python Security & More

Course Snapshot

www.triveratech.com

- **Course: TTPS4894: Python Security | Introduction to Python Programming for Security Analysts & Professionals**
- **Duration:** 4 days
- **Audience & Skill-Level:** This is an **introductory-level** course geared for Security professionals whose responsibilities include application security relative to Python applications.
- **Hands-on Learning:** This hands-on course combines engaging expert lessons, demos and group discussions with real-world, skills-focused machine-based labs and exercises. Student machines are required.
- **Delivery Options:** This course is available for **onsite private classroom presentation**, **live online virtual presentation**, or can be presented in a **blended learning format**. Please also ask about our **Self-Paced / Video / QuickSkills** or **Mini-Camp / Short Course** flexible delivery options.
- **Public Schedule:** This course has active dates on our live-online open enrollment **Public Schedule**.
- **Customizable:** This course agenda, topics and labs can be further adjusted to target your specific training skills objectives, tools and learning goals. Please ask for details.

Overview

Geared for experienced security professionals new to Python, Python Programming for Security Analysts & Professionals is practical, hands-on Python training course that leads the student from the basics of writing and running Python scripts to more advanced features such as file operations, regular expressions, working with binary data, and using the extensive functionality of Python modules. Extra emphasis is placed on features unique to Python, such as tuples, array slices, and output formatting.

This course is tailored specifically for Security Analysts and others who wish to use Python functionality for security-related tasks such as log manipulation or forensics. This course is essential for security professionals that are performing security reviews and audits of Python applications or are supporting development teams in implementing better defenses in Python.

Learning Objectives

This course combines expert lecture, real-world demonstrations and group discussions with machine-based practical labs and exercises. Throughout the hands-on course students will learn to write essential Python scripts using the most current and efficient skills and techniques.

Working in a hands-on learning environment, guided by our expert team, attendees will learn to:

- Create working Python scripts following best practices
- Use python data types appropriately
- Read and write files with both text and binary data
- Search and replace text with regular expressions
- Get familiar with the standard library and its work-saving modules
- Use lesser known but powerful Python data types
- Create "real-world", professional Python applications
- Work with dates, times, and calendars
- Know when to use collections such as lists, dictionaries, and sets
- Understand Pythonic features such as comprehensions and iterators
- Write robust code using exception handling
- Write Secure Python Applications
- Perform Log File Analysis
- Work with Security Filters, Packet Analysis and related Analytics
- Time Permitting / Bonus Content: Working with RESTful Services

Need different skills or topics? If your team requires different topics or tools, additional skills or custom approach, this course may be further adjusted to accommodate. We offer additional python, security, web development, data science, machine learning and other related topics that may be blended with this course for a track that best suits your goals. Our team will collaborate with you to target the course to focus on your specific learning objectives.

Audience & Pre-Requisites

This course is tailored specifically for Security Analysts and others new to Python, who wish to learn and use Python functionality for security-related tasks such as log manipulation or forensics. Students are required to have some basic programming experience and exposure prior to attending this course. Students should have basic development experience in any programming language, along with a working, user-level knowledge of Unix/Linux, Mac, or Windows.

Follow On Courses: Our Python tracks include a wide variety of follow-on courses and learning paths for leveraging Python for next-level web development, data science / machine learning, networking, task automation, security and other topics. Please see the attached **Python Training Suite** list of courses, or inquire for recommendations based on your specific role and goals.

Course Topics / Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We'll work with you to tune this course and level of coverage to target the skills you need most. Topics, agenda and labs are subject to change, and may adjust during live delivery based on audience interests, skill-level and participation.

- 1. An Overview of Python**
 - What is python?
 - Python Timeline
 - Advantages/Disadvantages of Python
 - Getting help with pydoc
 - 2. The Python Environment**
 - Starting Python
 - Using the interpreter
 - Running a Python script
 - Python scripts on Unix/Windows
 - Editors and IDEs
 - 3. Getting Started**
 - Using variables
 - Builtin functions
 - Strings
 - Numbers
 - Converting among types
 - Writing to the screen
 - Command line parameters
 - 4. Flow Control**
 - About flow control
 - White space
 - Conditional expressions
 - Relational and Boolean operators
 - While loops
 - Alternate loop exits
 - 5. Sequences**
 - About sequences
 - 6. Working with files**
 - File overview
 - Opening a text file
 - Reading a text file
 - Writing to a text file
 - Reading and writing raw (binary) data
 - Converting binary data with struct
 - 7. Dictionaries and Sets**
 - About dictionaries
 - Creating dictionaries
 - Iterating through a dictionary
 - About sets
 - Creating sets
 - Working with sets
 - 8. Functions**
 - Defining functions
 - Parameters
 - Global and local scope
 - Nested functions
 - 9. Sorting**
 - The sorted() function
 - Alternate keys
 - Lambda functions
 - Sorting collections
 - 10. Errors and Exception Handling**
 - Syntax errors
 - Exceptions
 - Using try/catch/else/finally
 - Handling multiple exceptions
 - Ignoring exceptions
 - 11. Modules and Packages**
 - The import statement
 - Module search path
 - Creating modules and Using packages
 - Function and Module aliases
 - 12. Working with Classes**
 - About o-o programming
 - Defining classes
 - Constructors
 - Methods
 - Instance data
 - Properties
 - Class methods and data
 - 13. Regular Expressions**
 - RE syntax overview
- Lists and list methods
 - Tuples
 - Indexing and slicing
 - Iterating through a sequence
 - Sequence functions, keywords, and operators
 - List comprehensions
 - Generator Expressions
 - Nested sequences
 - Returning values

- RE Objects
- Searching and matching
- Compilation flags
- Groups and special groups
- Replacing text
- Splitting strings

14. The standard library

- The sys module
- Launching external programs
- The string module
- Reading CSV data

15. Dates and times

- Working with dates and times
- Translating timestamps
- Parsing dates from text

16. Working with the file system

- Paths, directories, and filenames
- Checking for existence
- Permissions and other file attributes
- Walking directory trees
- Creating filters with fileinput
- Security and File Access

17. Network services

- Grabbing web content
- Detecting Malformed Input

18. Writing secure Python applications

- Parsing command-line options
- Getting help with pydoc
- Safely handling untrusted data
- Managing eval() permissions
- Potential insecure packages
- Embedding code snippets in Python
- Embedding authentication data in Python
- Potentially dangerous operations:
 - File access
 - Operating system access
 - Calls to external services
 - Called to external data sources
- Static analysis tools such as Bandit

19. Log File Analysis

- Raw log file manipulation
- Fail2Ban
- Customizing Fail2Ban with Python

20. Security Filters

- SQL-Injection Detection
- ModSecurity CRS filtering

21. Packet Analysis

- Packet Sniffing in Python

22. Analytics

- Security Logging and Analytics
- Attack Detection and Defense
- Python and Spark High-Level Overview

Bonus Content / Time Permitting

23. RESTful Web Services

- What is Flask?
- Developing a Flask Web service
- Mapping resources using URLs
- Mapping resources using HTTP Negotiating data content

24. Python application security

- OWASP 2021 Top Ten Overview
- Python Code Access Control
- Options for Protecting Data
- Injection and Python
- Python and Data Validation
- Python and XML Processing
- Python and Known Vulnerable Components
- Python and Serialization/Deserialization

For More Information

For more information about our dedicated training services, collaborative coaching services, courseware licensing options, public course schedule, training management services, partner programs, or to see our complete list of course offerings and special offers please visit us at www.triveratech.com, email Info@triveratech.com or call us toll free at **844-475-4559**.

TRIVERA TECHNOLOGIES • Collaborative IT Training, Coaching & Skills Development Solutions
www.triveratech.com • toll free +1-844-475-4559 • Info@triveratech.com • Twitter TriveraTech

ONSITE, ONLINE & BLENDED TRAINING SOLUTIONS • PUBLIC / OPEN ENROLLMENT COURSES • COURSEWARE LICENSING & DEVELOPMENT MENTORING • ASSESSMENTS • LEARNING PLAN DEVELOPMENT • SKILLS IMMERSION PROGRAMS / RESKILLING / NEW HIRE / BOOT CAMPS PARTNER & RESELLER PROGRAMS • CORPORATE TRAINING MANAGEMENT • VENDOR MANAGEMENT SERVICES

Trivera Technologies is a Woman-Owned Small-Business Firm

Explore Trivera's Ways to Learn!

