SEC673: Advanced Information Security Automation with Python

6 Day Program	36 CPEs	Laptop Required

You Will Learn

- New pyWars features, virtual environments, and VSCode
- How to use unit testing to evaluate code in development
- Python object-oriented coding
- Decorators
- Iterators
- Context managers
- Data descriptors
- Object attribute security and attacks
- Multi-threading
- Multi-processing
- How to understand and mitigate object serialization attacks
- The right way to do application logging
- Command line tool automation with PEXPECT
- Interpreter and object model attacks

When the security team has a problem it can't solve, it turns to you. You know how to code and you can develop tools to fill the gaps left by the existing technology. But although you can write a decent script to get the job done, maintaining that code is sometimes a burden. Every new feature seems like a complete rewrite. You need your code to run faster and spread the workload over multiple threads or even multiple processors. When a user experiences an error, you are left guessing what might have gone wrong because your application doesn't provide sufficient logging. You wish your applications had the features, maintainability, and ease of use of the most popular open-source cybersecurity projects. You are worried that you are developing a program with security vulnerabilities. Or, if you are not worried about that, maybe you should be. One thing is clear: you are ready to take your coding skills to the next level. SEC673 is the course for you!

SEC673 is designed as the logical progression point for students who have completed SEC573 Automating Information Security with Python, or for those who are already familiar with basic Python programming concepts. The course jumps immediately into advanced concepts. It looks at coding techniques used by popular open-source information security packages and how to apply them to our own Python cybersecurity projects. We'll learn from the best of them, spending the week making information security for our project, named SPF100, as easy to develop and maintain as that of the most popular cybersecurity projects. You'll discover how to organize your code and use advanced programming concepts to make your code faster, more efficient, and easier to maintain.

- The course will show you the proper way to develop custom classes for use in information security applications. You'll learn about such topics as:
- Making your packages installable for Package Installer for Python (PIP) for easy distribution and updates
- Building a custom data structure that fits your application for faster development
- Using advanced features like decorators, generators, and context managers to simplify code
- Making programs run faster with multi-threading and multi-processing
- Eliminating cascading errors by implementing unit tests so small changes don't become big errors
- Undertaking proper log generation and handling in Python applications in order to identify those "works for me" errors
- Implementing application automation and interaction so that you can move on to more important tasks

Following the precedent set by SEC573, this course makes heavy use of hands-on labs. The pyWars server is back with brand new functionality that can evaluate code and tell you where you've made mistakes. This unique and practical environment can guide you through developing complex programs. In addition to evaluating the answers you come up with, the pyWars server will assess how you solved the problems and demand that you solve them the right way. In other words, by forcing you to use the advanced features taught in this course, the server will help you learn the habits you need to produce more maintainable code. At the same time, the new environment will feed you complex problems in small consumable chunks so that you are not overwhelmed by the difficulty of the challenges presented and can focus on the important skills being taught.

If you understand the essential skills of Python and how to develop a simple information security tool, but are ready to be more productive and write better tools, SEC673 is the course for you.

Section Descriptions

SECTION 1: Python Package Essentials

The first course section jumps straight into pyWars. SEC573 alumni will quickly learn the new features and how they will be used in SEC673, while veteran coders who come straight to this course will be introduced to this amazing learning platform. You'll learn how developers use unit tests to evaluate their programs during the development process and prevent small changes in core function from having cascading affects in your applications. We'll deep dive into the Python package structure, and you'll learn how setup.py can be used to build a deployable package and how to handle common structural errors such as circular references.

TOPICS: Virtual Environment; Using an IDE; Unit Testing; Building Packages; PIP Installable Package; Understanding Package Imports; Absolute vs. Relative Imports; Circular References

SECTION 2: Python Objects

This course section will teach you to develop custom Python objects and data structures to support the needs of modern cybersecurity projects. We will build a model cybersecurity project called the Security Professionals Friend 100 (SPF100) that incorporates features found in popular cybersecurity packages such as Scapy and Volatility. You'll see how the right data structure can make your applications much easier to use and speed the development process. The section starts with a discussion on argument packing, unpacking, and how to pass arguments on to other functions. That discussion will set you up for success when we move into the principles of object-oriented coding and object inheritance, followed by a comprehensive discussion on object classes, scope, and inheritance that focuses on the real-world application of objects in modern Python projects. You'll learn to take advantage of existing data structure and extend build in object types like lists, integers, and dictionaries. Imagine having a data structure perfectly suited for your application, and then building it! Your code will be much cleaner and easier to support. You will learn how the magic dunder methods work and when to modify them in your programs. Finally, you'll learn how to use slicing in your own data types so you can pull the data you want from them.

TOPICS: Argument Packing; Objects; Inheritance Super; Inheriting and Extending Built-in Objects; The Magic Dunders; Slicing

SECTION 3: Python Objects (continued)

In this section we continue adding new features to SPF100 that make Python objects more versatile. You'll learn how to customize the behavior of objects when accessing their attributes, and how to resolve attribute naming conflicts by using name mangling. We will discuss attribute privacy, the security pitfalls associated with any developer trying to protect object attributes, and how to exploit them. This section will provide you with a firm understanding of how to perform error handling in your projects. We'll complete the session by adding custom iterators to our project that can process network packets in interesting ways.

TOPICS: Attribute Access; Executable Attributes; Name Mangling; Attribute Privacy; Object Comparison Operations; Advanced Exception Handling; Object Iteration; Object Instantiation

SECTION 4: Advanced Concepts

Knowing how to code is only part of the battle. When it comes to solving real-world cybersecurity problems, a bit more is required. Show us an information security professional who doesn't hate working with timestamps and time zones and we'll show you an information security professional who has never had to deal with timestamps and time zones. In this course section you will learn how to properly process and handle timestamps and solve problems that require knowledge of multiple time zones. You will learn how and when to use multi-processing and multi-threading to spread out the load and handle large amounts of data. We will continue to build on SPF100 and leverage Python features such as context managers in order to make the package as user-friendly as it is in other popular cybersecurity projects.

TOPICS: Dataclasses and NamedTuples; Timestamps and Time Zones; Concurrency; Serialization Attack and Mitigation; Context Managers

SECTION 5: Advanced Concepts (continued)

This course section will examine some of the most common struggles developers face when designing cyber tools. We will discuss how to automate command line tools that require interaction. This goes far beyond just running the program and capturing the output. We will talk about the ability to fully automate and interact with any command line. Next, we will add the ability to generate logs. You'll learn how to control the logs for other modules and configure applications so that you are alerted when critical events take place. We'll show you how you can use decorators to quickly add functionality to existing code with minimal changes to those programs. You will learn how to develop your own powerful decorators to improve any code base. We'll wrap up our discussion with a look at more security vulnerabilities that affect the Python interpreter and commonly used functions.

TOPICS: CLI Tool Automation; Logging; Decorators; Python Attacks

SECTION 6: Capture-the-Flag Challenge

In this capstone event, you will apply the skills you have mastered and the code you have developed throughout the course in a series of programming challenges. You will exploit vulnerable systems, built custom objects, decorators, and most of the other skills you have learned over the week.

Who Should Attend

- Security professionals who know how to code in Python and are ready to take their coding skills to the next level
- Tool developers who want to be able to publish installable and easy-to-use Python packages
- Network defenders who want to be able to extend the capability of popular Python packages to create new detection capabilities
- Security professionals who need their tools to run faster by adding multi-processing and multi-threading capabilities

Author Statement

"I've been overwhelmed by the popularity of the SEC573 course and the excitement about it. The most common feedback I get is 'We want MORE pyWars!' SEC673 is the answer to that call. And here's a bonus: What if while you are having all of that fun playing pyWars I could teach you to make applications run faster with multi-processing and multi-threading? What if you learned object-oriented coding, unit testing, how to have properly configured logging in your applications, and more? Well, that would be an awesome course.

The pyWars server has all-new capabilities that go beyond measuring whether or not you got the correct answer. Now it can assess the quality of the programs you are writing. I can feed you partially completed applications and have you complete the code to satisfy the learning objectives. You will go from writing simple modules and single file scripts to writing more complex and better organized Python packages. If you are ready to take your coding skills to the next level, come check out SEC673."

—Mark Baggett