# SEC573

**GPYC** Certification
Python Coder

www.giac.org/gpyc

## Automating Information Security with Python

**Six-Day Program**
**36 CPEs**
**Laptop Required**

### Who Should Attend

- Security professionals who want to learn how to develop Python applications
- Penetration testers who want to move from being a consumer of security tools to being the creator of security tools
- Technologists who need custom tools to test their infrastructure and who want to create those tools themselves

### You Will Be Able To

- Develop forensics tool to carve artifacts from forensics evidence for which no other tool exists or use third party modules for well-known artifacts to hidden evidence relevant to your investigations.
- Create defensive tools to automate the analysis of log file and network packets using hunt team techniques to track down attackers in your network. Implement custom whitelisting, blacklisting, signature detection, long tail and short tail analysis, and other data analysis techniques to find attacks overlooked by conventional methods.
- Write penetration testing tools including a several backdoors with features like process execution, upload and download payloads, port scanning and more. Build essential tools that evade antivirus software and allow you to establish that required foothold inside your target.
- Understand Python coding fundamentals required to automate common information security tasks. Language essentials like variables, loops, if then else, logic, file operations, command line arguments, debugging are all covered assuming no prerequisite knowledge.
- Tap into the wealth of existing Python modules to complete tasks using Regular Expressions, Database interactions with SQL, IP Networking, Exception handling, Interact with websites using Requests, Packet Analysis, Packet reassembly techniques and much more.

All security professionals, including Penetration Testers, Forensics Analysts, Network Defenders, Security Administrators, and Incident Responders, have one thing in common: CHANGE. Change is constant. Technology, threats, and tools are constantly evolving. If we don't evolve with them, we'll become ineffective and irrelevant, unable to provide the vital defenses our organizations increasingly require.

Maybe your chosen Operating System has a new feature that creates interesting forensics artifacts that would be invaluable for your investigation, if only you had a tool to access it. Often for new features and forensics artifacts, no such tool has yet been released. You could try moving your case forward without that evidence or hope that someone creates a tool before the case goes cold. **Or you can write a tool yourself.**

Perhaps an attacker bypassed your defenses and owned your network months ago. If existing tools were able to find the attack, you wouldn't be in this situation. You are bleeding sensitive data and the time-consuming manual process of finding and eradicating the attacker is costing you money and hurting your organization big time. The answer is simple if you have the skills: **Write a tool to automate your defenses.**

Finally, what do you do when "off-the-shelf" tools and exploits fall short? As a penetration tester you need to evolve as quickly as the threats you are paid to emulate, so the answer is simple, if you have the skills: **You write your own tool.**

Writing a tool is easier said than done, right? Not really. Python is a simple, user-friendly language that is designed to make automating tasks that security professionals perform quick and easy. Whether you are new to coding or have been coding for years, **SEC573: Automating Information Security with Python** will have you creating programs to make your job easier and make you more efficient. This self-paced class starts from the very beginning assuming you have no prior experience or knowledge of programming. We cover all of the essentials of the language up front. If you already know the essentials, you will find that the pyWars lab environment allows advanced developers to quickly accelerate to more advanced material in the class. The self-paced style of the class will meet you where you are to let you get the most out of the class. Beyond the essentials we discuss file analysis, packet analysis, forensics artifact carving, networking, database access, website access, process execution, exception handling, object-oriented coding and more.

This course is designed to give you the skills you need for tweaking, customizing, or outright developing your own tools. We put you on the path of creating your own tools, empowering you in automating the daily routine of today's information security professional, and in achieving more value in less time. Again and again, organizations serious about security emphasize their need for skilled tool builders. There is a huge demand for people who can understand a problem and then rapidly develop prototype code to attack or defend against it. **Join us and learn Python in-depth and fully weaponized.**

SANS
www.sans.org/SEC573

SANS Technology Institute
www.sans.edu

## 573.1 HANDS ON: Essentials Workshop with pyWars

The course begins with a brief introduction to Python and the pyWars Capture-the-Flag game. We set the stage for students to learn at their own pace in the 100% hands-on pyWars lab environment. As more advanced students take on Python-based Capture-the-Flag challenges, students who are new to programming will start from the very beginning with Python essentials.

**Topics:** **Python Syntax; Variables; Math Operators; Strings; Functions; Modules; Control Statements; Introspection**

## 573.2 HANDS ON: Essentials Workshop with MORE pyWars

You will never learn to program by staring at PowerPoint slides. The second day continues the hands-on, lab-centric approach established on day one. This section covers data structures and more detailed programming concepts. Next, we focus on invaluable tips and tricks to make you a better Python programmer and how to debug your code.

**Topics:** **Lists; Loops; Tuples; Dictionaries; The Python Debugger; Coding Tips, Tricks, and Shortcuts; System Arguments; ArgParser Module**

## 573.3 HANDS ON: Defensive Python

Day three includes in-depth coverage about how defenders can use Python automation as we cover Python modules and techniques that everyone can use. Forensicators and offensive security professionals will also learn essential skills they will apply to their craft. We will play the role of a network defender who needs to find the attackers on their network. We will discuss how to analyze network logs and packets to discover where the attackers are coming from and what they are doing. We will build scripts to empower continuous monitoring and disrupt the attackers before they exfiltrate your data.

**Topics:** **File Operations; Python Sets; Regular Expressions; Log Parsing; Data Analysis tools and techniques; Long Tail/Short Tail Analysis; Geolocation Acquisition; Blacklists and Whitelists; Packet Analysis; Packet Reassembly; Payload Extraction**

SANS

### *SEC573 Training Formats*

*(subject to change)*

**Live Training**
www.sans.org/security-training/by-location/all

**Summit Events**
www.sans.org/summit

**Mentor Training**
www.sans.org/mentor

**Private Training**
www.sans.org/onsite

## 573.4 HANDS ON: Forensics Python

On day four we will play the role of a forensics analyst who has to carve evidence from artifacts when no tool exists to do so. Even if you don't do forensics you will find that these skills covered on day four are foundational to every security role. We will discuss the process required to carve binary images, find appropriate data of interest in them, and extract that data. Once you have the artifact isolated, there is more analysis to be done. You will learn how to extract metadata from image files. Then we will discuss techniques for finding artifacts in other locations such as SQL databases and interacting with web pages.

**Topics:** **Acquiring Images from Disk, Memory, and the Network; File Carving; The STRUCT Module; Raw Network Sockets and Protocols; Image Forensics and PIL; SQL Queries; HTTP Communications with Python Built-In Libraries; Web Communications with the Requests Module**

## 573.5 HANDS ON: Offensive Python

On day five we play the role of penetration testers whose normal tricks have failed. Their attempts to establish a foothold have been stopped by modern defenses. To bypass these defenses, you will build an agent to give you access to a remote system. Similar agents can be used for Incident response or systems administration, but our focus will be on offensive operations.

**Topics:** **Network Socket Operations; Exception Handling; Process Execution; Blocking and Non-blocking Sockets; Asynchronous Operations; The Select Module; Python Objects; Argument Packing and Unpacking**

## 573.6 HANDS ON: Capture the Flag

In this final section you will be placed on a team with other students. Working as a team, you will apply the skills you have mastered in a series of programming challenges. Participants will exercise the skills and code they have developed over the previous five days as they exploit vulnerable systems, break encryption cyphers, analyze packets, parse logs, and automate code execution on remote systems. Test your skills! Prove your might!

"Best class ever! After just 2 days I'm getting comfortable with the nuances of Python. I never thought that would happen." -Jay Wilson, Navient