# Book Review: Violent Python

Review by Andrew Johnson OSCE, OSCP, GWAPT, GPEN, et al

As stated in its tagline, Violent Python is A Cookbook for Hackers, Forensic Analysts, Penetration Testers, and Security Engineers. This is a relatively broad scope and demonstrates how Python can be used to automate and assist with tasks across a variety of diverse InfoSec disciplines. However, breadth does not preclude depth in this case; the exercises build up to a fairly advanced level. Violent Python is authored primarily by TJ O’Connor, with Rob Frost contributing a chapter on Web Reconnaissance, and Mark Baggett acting as the Technical Editor. A quick glance at their collective credentials and experience undoubtedly creates high expectations for this title.

For those unfamiliar with cookbook-style resources, the contents are made up of dozens of short, self-contained “recipes.” The objective is not to comprehensively teach Python from the ground-up, but rather present scripts that focus on a specific task. The end result is that the book demonstrates how powerful just a few dozen lines of Python code can be (even the longest of recipes rarely exceed 100 lines). However, while the aim is not to teach Python programming in general, useful tips and tricks will surely be acquired simply by working through the exercises. The recipes were created in a modular fashion, with code reusability in mind, and they can easily be incorporated into larger projects. Let’s take a closer look.

Discuss in Forums {mos_smf_discuss:Book Reviews}

Before diving in, it should be mentioned that the files containing the source code for all recipes in the book are located on its Syngress Homepage. It is recommended to write the code from scratch as the book is read to greatly facilitate learning and retention, but having the code quickly and easily accessible is a nice perk (should it ever be needed).

The book’s contents are broken down as follows:

• Chapter 1: Introduction

• Chapter 2: Penetration Testing with Python

• Chapter 3: Forensic Investigations with Python

• Chapter 4: Network Traffic Analysis with Python

• Chapter 5: Wireless Mayhem with Python

• Chapter 6: Web Recon With Python

• Chapter 7: Antivirus Evasion with Python

Dozens of recipes are packed into the book’s 288 pages, and readers will likely be surprised at the types of results that can be so easily achieved with Python. While space constraints prevent every recipe from being enumerated in detail, some of the more interesting recipes that are covered throughout the course of the book include:

• Developing online and offline password crackers

• Interacting with Nmap and Metasploit

• Recreating Conficker

• Delivering an exploit for a stack-based buffer overflow

• Pillaging iTunes backups, the Windows registry, and SQLite databases

• Correlating network traffic to physical locations and creating Google Earth maps

• Building an SSH Botnet

• IDS and AV evasion

• Parsing metadata, web sites, and Tweets

• Creating social engineering email campaigns

• Parsing and logging wireless traffic

• Orchestrating Bluetooth-based attacks

• Hijacking a UAV

The introduction discusses how to setup a Python development environment and serves as a crash-course in the language itself. This will help the reader get up and running, and may serve as a useful refresher for those who have dabbled in the language previously. However, it may be inadequate for someone who has never touched the language before. Note: This is not a criticism of the book, as it would be quite unfair to expect Python programming to be covered thoroughly in a handful of pages when entire books have been dedicated to the subject.

Additionally, many of the explanations in the book discuss what actions a particular block of code performs, as opposed to providing a line-by-line breakdown. Being able to read common Python statements will allow the reader to more quickly understand and assimilate the core material. Those with little-to-no Python experience may find themselves better prepared to journey through this title by taking a brief detour and starting with Google’s free two-day course and/or Learn Python the Hard Way.

That being said, readers definitely do not need to possess expert-level Python knowledge to appreciate the material. Those who prefer the trial-by-fire approach to learning can certainly dive right in and learn general Python along with the InfoSec topics. The code is clean and well-written, and the narrative is presented in a casual manner that keeps the use of jargon to a minimum. The majority of the contents should be easily accessible to novices that may lack experience or have a weak grasp on development lingo. At the same time, the amount of unique and interesting material included should appeal to more experienced Python developers as well.

The authors went out of their way to make the recipes meaningful, and this significantly increases the value and enjoyment delivered by this title. The book is far from a collection of scripts that have been copied-and-pasted into relevant chapters. Instead, the recipes are often prefaced with interesting and/or entertaining background information. Setting up recipes in this manner creates scenarios with clearly defined objectives, which range from recreating notorious attacks and solving investigative dilemmas, to executing an attack just because it’s cool. This extra effort was an unexpected bonus, and the educational value of the historical trivia occasionally rivals that of the recipe itself.

Some recipes are quite interesting but appear impractical on the surface. The majority of InfoSec professionals likely have little need to run an SSH botnet or take down a UAV. However, the core concepts that are taught, such as being able to interact with a large number of hosts via SSH, and injecting packets into wireless traffic, respectively, could be widely applicable during a penetration test or other information security activities. The use of such scenarios for the recipes will keep readers engage as they progress through the book.

It’s important to note that the focus is always exclusively on Python. This is unsurprising given the nature of the book, but it’s worth emphasizing because some recipes require outside knowledge to truly understand what is going on behind-the-scenes. The exploitation recipe is a perfect example of this. This recipe shows how to deliver an exploit via Python (which in itself can serve as a useful template that can be reused for developing network-based proof-of-concept exploits). While the recipe begins with a brief overview of Stack-Based Buffer Overflows, it does not discuss identifying the vulnerability, assembly language, x86 memory management, or creating the shellcode. Such topics are obviously outside the scope of this book.

It can be difficult accommodate all readers completely, since knowledge and skill levels will vary significantly across the audience. In order to address potential gaps in external knowledge, tips and tricks along with common gotchas and mistakes are addressed in sidebars. Each chapter also contains numerous references that serve as excellent resources for further study and research.

In conclusion, Violent Python is an excellent resource that develops and enhances a diverse set of security-related Python skills. The book also serves as a great reference, where recipes could be quickly implemented to address specific issues as they arise. Readers who are actively working in an InfoSec position will likely be able to put their newly acquired skills to use immediately, and those looking to break into the field will acquire skills that differentiate themselves from others who are dependent on prebuilt tools. This title is highly recommended for anyone who wants to improve his or her Python skills within the InfoSec realm.

Andrew Johnson (OSCE, OSCP, GWAPT, GPEN, GCIA, GCIH, GSEC, CISSP, CEH, eCPPT, OSWP, CCNA:S, MCSE:S, et al) has over a decade of experience in information technology and security. He has provided information security services, including penetration testing, social engineering, and risk management to over one hundred financial institutions, businesses, and other organizations across the country. He has also performed information security management for the US operations of a global financial services ASP, servicing over 650 financial institutions and safeguarding more than 40 million sensitive records. He currently delivers penetration testing, training, and security consulting services to an international client base, ranging from niche web start-ups to Fortune 500 companies. Andrew is a perpetual learner and enjoys sharing knowledge with others, is a SANS Advisory Board Member and Exam Question Writer, and is a member of OWASP, InfraGard, and American MENSA. His personal security blog is at www.infosiege.net, and he can be followed @infosiege.