Coding for Penetration Testers
Coding for Penetration Testers
Building Better Tools

Jason Andress
Ryan Linn
Contents

Foreword...................................................................................................................xi
About the Authors .................................................................................................. xiii
About the Technical Editor ................................................................................... xv
Acknowledgments................................................................................................. xvii
Chapter 0: Introduction .......................................................................................... xix

CHAPTER 1 Introduction to command shell scripting ............... 1

On Shell Scripting................................................................................................. 1
    What is a shell?............................................................................................... 2
    What is a script?............................................................................................ 3
    Shell scripts................................................................................................. 3
    Where shell scripting is useful..................................................................... 4

UNIX, Linux, and OS X shell scripting ............................................................. 5
    Shell availability and choices...................................................................... 5
    Working with shells...................................................................................... 7

Bash basics ........................................................................................................... 8
    Hello World................................................................................................. 8
    Variables...................................................................................................... 10
    Arguments.................................................................................................... 10
    Control statements...................................................................................... 11

Putting it all together with bash........................................................................ 15
    Adding /dev/tcp/ support to bash.............................................................. 15
    Building a port scanner with bash.............................................................. 16
    Improving the script................................................................................... 18

Windows scripting............................................................................................. 18
    Shell availability and choices...................................................................... 18
    Command.com and CMD.exe ...................................................................... 18
    PowerShell................................................................................................. 19
    Cygwin......................................................................................................... 20
    Other shells................................................................................................. 21

PowerShell basics.............................................................................................. 21
    Hello World............................................................................................... 22
    Variables...................................................................................................... 23
    Arguments.................................................................................................... 25
    Control statements...................................................................................... 26
    Conditionals................................................................................................. 27
    Looping......................................................................................................... 28

Putting it all together with PowerShell............................................................ 29
    Building a port scanner with PowerShell................................................... 30
    Improving the script................................................................................... 32

Summary............................................................................................................... 32

Endnotes............................................................................................................... 33
## CHAPTER 2  Introduction to Python .................................................35
What is Python? ...........................................................................35
  Where do we get Python? ......................................................36
Where is Python useful? .............................................................36
  Multiplatform scripting..........................................................36
  Network scripting ...................................................................36
  Extensive modules ...............................................................37
  Reusable code that is easy to create ......................................37
Python basics ...............................................................................38
  Getting started........................................................................38
  Variables ................................................................................39
  Modules ..................................................................................40
  Arguments ..............................................................................41
  Lists ......................................................................................44
  Dictionaries ..........................................................................46
  Control statements ...............................................................51
  Functions ...............................................................................52
File manipulation.........................................................................54
  Exception handling ...............................................................55
Network communications............................................................57
  Client communications ........................................................57
  Server communications .......................................................59
  Scapy ...................................................................................62
Summary ......................................................................................68
Endnotes ......................................................................................68

## CHAPTER 3  Introduction to Perl .................................................69
Where Perl is useful ....................................................................69
  Handling text ..........................................................................70
  Gluing applications together ...............................................70
Working with Perl .......................................................................71
  Editing tools ..........................................................................71
  Extending Perl scripts .........................................................72
  GUIs in Perl ..........................................................................73
Perl basics ....................................................................................73
  Hello World ..........................................................................73
  Variables ...............................................................................75
  Shell commands .....................................................................76
  Arguments ..............................................................................79
  Control statements ...............................................................79
  Regular expressions ............................................................85
  File input and output ...........................................................87
Putting it all together.................................................................91
CHAPTER 7 Scanner scripting................................. 181
Working with scanning tools..................................... 181
Netcat ................................................................. 181
Nmap ................................................................. 182
Nessus/OpenVAS .................................................. 182
Netcat ................................................................. 183
  Implementations of Netcat ...................................... 183
  Simple Netcat usage ............................................ 184
  Building a Web server with Netcat .......................... 185
  Transferring files with Netcat ................................ 187
Nmap ................................................................. 191
  Working with service probes in Nmap ....................... 191
  The Nmap scripting engine ................................... 194
  Building Nmap NSE files .................................... 194
Nessus/OpenVAS .................................................. 196
  NASL in Nessus and OpenVAS ............................... 196
  Nessus attack scripting language (NASL) .................. 196
Summary ............................................................. 199
Endnotes ................................................................... 200

CHAPTER 8 Information gathering ......................... 201
Information gathering for penetration testing ............. 201
Sources of information .......................................... 202
Patterns in information ......................................... 202
Metadata ................................................................... 203
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux network information gathering</td>
<td>261</td>
</tr>
<tr>
<td>Scripting Metasploit Meterpreter</td>
<td>262</td>
</tr>
<tr>
<td>Getting a shell</td>
<td>262</td>
</tr>
<tr>
<td>Building a basic script</td>
<td>264</td>
</tr>
<tr>
<td>Executing the script</td>
<td>269</td>
</tr>
<tr>
<td>Database post-exploitation</td>
<td>270</td>
</tr>
<tr>
<td>What is SQL injection?</td>
<td>270</td>
</tr>
<tr>
<td>MySQL</td>
<td>271</td>
</tr>
<tr>
<td>SQL injection on Microsoft SQL Server</td>
<td>278</td>
</tr>
<tr>
<td>Summary</td>
<td>280</td>
</tr>
<tr>
<td>Appendix</td>
<td>283</td>
</tr>
<tr>
<td>Index</td>
<td>285</td>
</tr>
</tbody>
</table>
Foreword

My Dear Reader,

This wretched war, the gravest threat humankind has ever faced, is not going well at all. We have suffered major setbacks, as our ruthless adversary has conquered vast territories, leaving little ground controlled by our ragtag band of rebels. Our few surviving generals blame the lack of skills in our fighting forces, allowing the enemy to rout us in every hard-fought battle. Our situation is dire.

Historians have traced this impossibly sad state of affairs to some crucial mistakes we made collectively in the 2012–2015 time frame. We had spent the prior 30 years building ever more powerful networked machines, including PCs, smartphones, and industrial control systems, all interconnected on that blasted Internet. At first, before 2012, the machines were our servants, mindless systems processing transactions, scurrying about vacuuming our floors, and otherwise making life more pleasant for humans. Then, in 2012, Moore’s relentless law kicked things into maximum overdrive. Within a decade, the machines had become sentient, matching the smartest humans on the planet. They quickly became our most trusted advisors and friends. We should have seen the warning signs and used that precious time to develop our skills. Instead, we stupidly let ourselves atrophy. As they surpassed humans, the machines began to view us as pets, but we rejected their control. Soon, they came to the conclusion that humans were a disease, a cancer of this planet, and they viewed themselves as the cure, tirelessly working for our eradication. The war began.

We could have stopped them, I tell you, if only we had enough people with scripting and coding skills.

Through an astonishing scientific breakthrough, our physicists have managed to figure out a way to transmit this message back in time to you. I have been tasked by the Human Ruling Council to ask . . . no, beg you to read this book and master its skills so that you can turn the tide of history itself. In these pages, you will learn how to wield control of computer systems through writing scripts and code in a variety of the most important languages today: Python, Ruby, PowerShell, and more. You’ll also learn how to apply various coding concepts to extend the capabilities of some of the most powerful free security scanners and tools. The book covers these topics from a penetration tester’s perspective, showing you how to find and exploit security flaws in the exciting and rapidly growing field of information security. What’s more, using the automation available in these powerful scripting languages and tools, you’ll be able to improve defenses throughout enterprises of any scale, from small mom-and-pop shops to large multinationals. These skills will help both security professionals and general IT practitioners do their jobs more effectively. The book is eminently practical, showing you how to get real stuff done in these scripting languages. That’s your immediate payoff.

But its usefulness in improving your skills and career isn’t the only reason to read the book. I won’t mince words — our very survival as a species is inherently linked
to your mastering the knowledge of this book. We need you to learn script writing to keep the machines in check over your coming decade so that you can avoid our sad fate. I implore you to learn it and live it, for your sake and for that of future generations. What are you waiting for? Help us, Dear Reader. You’re our only hope!

—Ed Skoudis
SANS Instructor and
Co-Founder of Counter Hack Challenges
June 10, 2011
About the Authors

Jason Andress (ISSAP, CISSP, GPEN, CEH) is a seasoned security professional with a depth of experience in both the academic and business worlds. Presently he carries out information security oversight duties, performing penetration testing, risk assessment, and compliance functions to ensure that critical assets are protected.

Jason has taught undergraduate and graduate security courses since 2005 and holds a doctorate in computer science, researching in the area of data protection. He has authored several publications and books, writing on topics including data security, network security, penetration testing, and digital forensics.

Ryan Linn is a penetration tester, an author, a developer, and an educator. He comes from a systems administration and Web application development background, with many years of information technology (IT) security experience.

Ryan currently works as a full-time penetration tester and is a regular contributor to open source projects including Metasploit, The Browser Exploitation Framework, and the Dradis Framework. He has spoken at numerous security conferences and events, including ISSA, DEF CON, SecTor, and Carolina Con. As the twelfth step of his WoW addiction recovery program, he has gained numerous certifications, including the OSCE, GPEN, and GWAPT.
About the Technical Editor

**Russ Rogers** (CISSP, CISM, IAM, IEM, Hon. Sc.D.), author of the popular *Hacking a Terror Network: The Silent Threat of Covert Channels* (Syngress, ISBN: 978-1-928994-98-5), co-author of multiple books, including the best-selling *Stealing the Network: How to Own a Continent* (Syngress, ISBN 978-1-931836-05-0) and *Network Security Evaluation Using the NSA IEM* (Syngress, ISBN: 978-1-59749-035-1), and former editor-in-chief of *The Security Journal*, is currently a penetration tester for a federal agency and the co-founder and chief executive officer of Peak Security, Inc., a veteran-owned small business based in Colorado Springs, CO. Russ has been involved in information technology since 1980 and has spent the past 20 years working as both an IT and InfoSec consultant. Russ has worked with the U.S. Air Force (USAF), National Security Agency (NSA), Defense Information Systems Agency (DISA), and other federal agencies. He is a globally renowned security expert, speaker, and author who has presented at conferences around the world in Amsterdam, Tokyo, Singapore, São Paulo, Abu Dhabi, and cities all over the United States.

Russ has an honorary doctorate of science in information technology from the University of Advancing Technology, a master’s degree in computer systems management from the University of Maryland, a bachelor of science degree in computer information systems from the University of Maryland, and an associate’s degree in applied communications technology from the Community College of the Air Force. Russ is currently pursuing a bachelor of science degree in electrical engineering from the University of Colorado at Colorado Springs. He is a member of ISSA and (ISC)²® (CISSP). Russ also teaches at and fills the role of professor of network security for the University of Advancing Technology (www.uat.edu).
Acknowledgments

Jason Andress
Thanks to my family for persevering through another book project and for putting up with me. Additionally, thanks to Ryan for being a great coauthor and to Russ for handling the tech editing. Last, but certainly not least, thanks to Matt for keeping everything on the rails at Syngress. You work way too hard and deserve an extra half-ration of gruel.

Ryan Linn
I would like to thank Jason, Russ, Audrey, Matt, and the folks at Syngress for all of their contributions and assistance through this process. Thanks to Heather for her support; without your aid, I’m not sure this book would have ever happened. Thanks to Ed, Kevin, HD, JCrán, Egypt, Wade, Don, CG, JJ, Brian, and the other mentors that have helped me along my security career; all of your generosity and patience have helped me grow to a point where I can share with others. Thanks to the security community for being a generally awesome group of people who are willing to share knowledge and skills. To L0pht, Offsec, Corelan, and other security researchers of today, thank you for the knowledge sharing and explanations that are always pushing folks to learn more; thanks for bringing your knowledge to the rest of the community so that we can all learn.

Last, but not least, I would like to thank my family. Thank you all for your support, guidance, cheering, and, oh yeah, making sure I made it this far without being eaten by wolves.
Introduction

INFORMATION IN THIS CHAPTER:
- Book Overview and Key Learning Points
- Book Audience
- How This Book Is Organized

BOOK OVERVIEW AND KEY LEARNING POINTS
What sets a good penetration tester apart from an average one is the ability to adapt to the ever-changing landscape within which we live. One aspect of this adaptability is the skill to build, extend, and manipulate scripts and applications encountered in the field. Whether tools already exist to accomplish a task, or one needs to be built to take advantage of a new vulnerability, the ability to build and extend tools in a variety of scripting languages is important. Each of the first five chapters of this resource delves into a different scripting language that we may encounter while performing penetration tests.

Through investigating the core aspects of each language, either on Microsoft platforms, or on Linux platforms such as BackTrack 5, each chapter brings to light the power and strengths of each language. We will use these strengths to build a series of scripts to help us understand the intricacies of each language, and in most cases develop a basic tool that we can use and extend while penetration testing. Whether it is through shell scripting, Python, Perl, Ruby, or PHP, we will cover the basics of each language and discuss topics such as output handling, loops and control statements, networking, and command execution.

Once the core language concepts have been covered, Coding for Penetration Testers tackles the core tasks of penetration testing. While covering scanner scripting and information gathering, we will discuss tools such as Nmap and Nessus and use the scripting languages behind them to extend the capabilities of both tools. Information gathering is one of the first and most important steps of a penetration test. We don’t know what we’re attacking until we do the initial research. Chapter 8 investigates how to automate information gathering tasks to be more effective and to have repeatable results.

Once we’ve gathered the information, we’re ready to begin the offensive. Through looking at Python as an exploit delivery tool, we will discuss the basics of exploit development. Walking through building a working exploit, Chapter 9 takes us through each step of the process, from creating a Proof of Concept (POC) to creating a repeatable and extendable exploit within the Metasploit Framework.

xix
Not to leave the Web applications out of the mix, Chapter 9 covers a variety of Web attack methodologies and looks at how to script these attacks to be more effective. Although we’ve successfully penetrated a system, that doesn’t mean we’re done. Chapter 10 brings in the testing we need to perform, with a discussion of post-exploitation tasks under Windows, Linux, and Web applications. By understanding how to query systems, expand our access, and mine systems for data, we will be able to extend our reach from one system to the network or database. By the end of the book, not only will we have covered the core languages encountered while penetration testing, but we also will have built and investigated real-world scenarios for how to use these languages, tools, techniques, and concepts.

BOOK AUDIENCE
This book will be a valuable resource to those involved in penetration testing activities, as well as security professionals and network and system administrators. The information provided on scripting languages and attacks can also be used to assist in engineering better and more efficient defenses.

Those in development positions will find this information useful as well, from the standpoint of developing better tools for their organizations. The concepts discussed in this book can be used to learn the basic concepts of several scripting languages, as well as working through the application of these languages in building practical tools.

In order to get the most out of this resource, some knowledge or experience is required. We will be going over networking, advanced Windows commands, and Web and application exploitation, so individuals need to have either a basic understanding of these tasks to get started, or a desire to learn those things alongside this text. In many cases, we point out resources to accompany the concepts that are being discussed, so don’t be intimidated. The desire to learn and understand new and progressive concepts is essential for a good penetration tester.

HOW THIS BOOK IS ORGANIZED
Due to the content and organization of the topics in this book, it is not necessary to read it from front to back. Chapter 1 through Chapter 6 and the Appendix cover language concepts, while the final chapters cover applications of these languages. When we refer to information in other chapters in the book, we include references to point to the chapter where the information can be found. The following descriptions provide an overview of the contents of each chapter:

Chapter 1: Introduction to command shell scripting
In this chapter, we talk about shells, in the sense of the text-based interfaces we use to communicate with operating systems. UNIX, Linux, and OS X, as well as
most UNIX-like operating systems, tend to work on the same general principles for purposes of shell scripting, and make use of many common programming concepts such as data structures, variables, control statements, if-then clauses, and while loops. In Microsoft operating systems, we can find many similar shell scripting tools as well. In Windows, we can carry out commands and write scripts using the generic shells command.com and CMD.exe, the PowerShell shell, and add-on tools such as Cygwin to give us access to bash on Windows, just to name a few.

Chapter 2: Introduction to Python

In this chapter, we dig into the Python language and investigate different types of network scripts, subnetting, and file manipulation. These topics are investigated through practical examples that we will encounter while penetration testing. We look at how to build scripts to communicate with Web servers, determine what Web servers may be hiding, and even investigate how to send our data without it being seen by network tools. These examples and more provide practical, real-world scenarios for when and how to use the Python language. While we’re building tools that we can extend for our scripting toolkit, we investigate all the Python basics from data structures, to control statements, to interacting with the shell, and manipulating strings and files.

Chapter 3: Introduction to Perl

In this chapter, we examine the Perl language. We can use Perl to process data and merge data together from disparate sources, a common function in the penetration testing world with its many tools. Scripting in Perl follows most of the standard conventions we can find in other scripting or programming languages and can make use of various data structures, such as variables and arrays to store data in our scripts, arguments, control statements such as loops and conditionals, as well as regular expressions, file I/O, and many of the other standard programming language features.

Chapter 4: Introduction to Ruby

Ruby is a flexible programming language that has taken the better parts of Perl, Python, and many others to create a language that is both powerful and easy to read. In this chapter, we take a look at the powerful object-oriented approach to Ruby. Using Ruby to convert data between hex, binary, and plaintext data, this chapter looks at the details of network and file manipulation, building new classes, manipulating databases, and even building basic network servers. Through these examples, we explore the critical aspects of the Ruby language, and develop the skills to aid in real-world testing scenarios.
Chapter 5: Introduction to Web scripting with PHP

Chapter 5 dives into the world of PHP and Web scripting. Through building basic Web applications with the HyperText Markup Language (HTML) and PHP, we can build dynamic Web pages that take advantage of file manipulation, databases, and even issuing system calls. This chapter focuses on some of the basics and works up to create Web shells that we can use in the field. While working through examples of file manipulation, command execution, loops, and data structures, this chapter walks through the core concepts of PHP that we will need to understand while testing, as well as helps us to create tools that we can use, extend, and incorporate into more sophisticated tools.

Chapter 6: Manipulating windows with PowerShell

This chapter delves deeper into the capabilities of PowerShell, which can be very handy in certain penetration testing situations. PowerShell has access to all the functionality of .NET and can give us capabilities that we might not otherwise have in such an environment without needing to upload tools to the system. We go over execution policies, taking control of the processes on the system, interfacing with the event logs, tweaking the Registry, and more, all through the tools provided by the operating system.

Chapter 7: Scanner scripting

This chapter covers the use of Netcat, Nmap, and Nessus/OpenVAS, and what we can do with them through scripting languages. We talk about automating Netcat through shell scripts, in order to allow us to send files, run simple network services, and forward ports, altering or adding to the behavior of Nmap, customization of Nessus and OpenVAS through the use of the Nessus Attack Scripting Language (NASL), and several other similar tasks.

Chapter 8: Information gathering

In this chapter, we look into information gathering and how it can be of great use to us in the course of a penetration test. We talk about automating searches with Google, parsing text and automating Web interaction with Perl, and finding and working with the metadata stored in documents, files, visual media, and other such structures intended for digital storage of information. We also look at the various tools we might want to use to search for and sift through such data once we have it.

Chapter 9: Exploitation scripting

The ability to use and build exploits is what sets penetration tests apart from vulnerability scans. This chapter works through building a simple exploit from scratch using Python. Once Python has squeezed the application into submission
and returned us a shell, we move to Ruby to make it repeatable and more powerful by converting it to a Metasploit module. After exploring the world of binary exploitation, this chapter moves into Web application testing and investigates how to deal with Remote File Inclusion (RFI) vulnerabilities and Cross-Site Scripting (XSS) vulnerabilities. This chapter walks through using the RFI vulnerabilities and leveraging the shells that were created in Chapter 5 to go from Web vulnerability to command line, and ends with building additional scripts to steal data using XSS.

Chapter 10: Post-Exploitation scripting
Once the shells come back, we have a bit more work to do. Chapter 10 discusses what happens after we’ve gotten in. Working from information gathering under Windows, and moving toward maintaining access through creating users using the Windows command line, this chapter looks at how to query and manipulate Windows systems from the command line. Once we’ve gotten the hang of it, Chapter 10 works to convert that hard work into a Meterpreter script using Ruby where we can easily run these commands through Metasploit.

After exploiting a Web application, we don’t want the database to feel left out, so this chapter ends with manipulating SQL injection vulnerabilities to gain access to applications, dump data, and even get a shell. While looking at capabilities of MySQL and Microsoft SQL Server, this chapter helps develop post-exploitation abilities that will start a basic Web application tester on the road to becoming an advanced one.

Appendix: Subnetting and CIDR addresses
The Appendix fills in information about subnetting for those who want to know more after the subnet calculator example from Chapter 2. The Appendix covers the basics of netmask calculation, Classless Inter-Domain Routing (CIDR) addressing, and the relationship among netmasks, IP addresses, broadcast addresses, and network addresses. Combined with the skills in Chapter 2, this knowledge should make subnetting easy.

CONCLUSION
Throughout the chapters in this book, our goal has been to share the basics of many of the languages we encounter every day while pen testing, by not just explaining the syntax, but also using those skills to build usable tools along the way. We hope this book will become a reference for people who are learning the trade, as well as those who are established. Working with multiple languages on a daily basis can be difficult, so we hope that as we all grow, this book will still prove to be a valuable reference on a regular basis.
As you read through this reference, you will find that many tools can be critical while performing a penetration test, but problematic when used without permission. Please use these tools responsibly and only on systems where you have permission. We have tried to facilitate this by framing most of the tools in the context of exercises that can be performed on a local BackTrack 5 installation or a local Windows virtual machine.

We are happy to have the opportunity to share our knowledge and enjoyment of the art of penetration testing with you. We hope you enjoy reading the following chapters as much as we enjoyed putting them together. We wish you good luck and many shells.