Cyber Operations

Building, Defending, and Attacking Modern Computer Networks

—

Second Edition

—

Mike O'Leary

Apress®

Cyber Operations

Building, Defending, and Attacking Modern Computer Networks

Second Edition

Mike O'Leary

Apress®

Cyber Operations: Building, Defending, and Attacking Modern Computer Networks

Mike O'Leary
Towson, MD, USA

ISBN-13 (pbk): 978-1-4842-4293-3 ISBN-13 (electronic): 978-1-4842-4294-0
https://doi.org/10.1007/978-1-4842-4294-0

Library of Congress Control Number: 2019933305

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Susan McDermott
Development Editor: Laura Berendson
Coordinating Editor: Rita Fernando

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit http://www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484242933. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

Dedicated to all of the security professionals
who volunteer their time to work with students.

Table of Contents

About the Author

Mike O'Leary is a professor at Towson University and was the founding director of the School of Emerging Technologies. He developed and teaches hands-on capstone courses in computer security for both undergraduate and graduate students. He has coached the Towson University Cyber Defense team to the finals of the National Collegiate Cyber Defense Competition in 2010, 2012, and 2014.

About the Technical Reviewer

Dr. Jacob G. Oakley spent over seven years in the U.S. Marines and was one of the founding members of the operational arm of Marine Corps Forces Cyberspace Command at NSA, Ft. Meade, Maryland, leaving that unit as the senior Marine Corps operator and a division technical lead. After his enlistment, he wrote and taught an advanced computer operations course, eventually returning back to mission support at Ft. Meade. He later left government contracting to do threat emulation and red teaming at a private company for commercial clients, serving as principal penetration tester and director of penetration testing and cyber operations. He is currently working as a Cyber SME for a government customer. He completed his doctorate in IT at Towson University researching and developing offensive cyber security methods and is the author of *Professional Red Teaming: Conducting Successful Cybersecurity Engagements* (Apress, 2019).

Acknowledgments

I would like to thank the students who have gone through my class over the years - this book would not exist without you. I hope to see you back!

Special thanks go to Jacob Oakley for his time and insight as technical reviewer.

I would also like to thank the members of the Apress team, including Rita Fernando and Susan McDermott, who have provided wonderful assistance over the two years it has taken to write this book.

I can't thank my family enough for giving me the time and the support to write this.

Introduction

How do you set up, defend, and attack computer networks? This book is a gentle introduction to cyber operations for a reader with a working knowledge of Windows and Linux operating systems and basic TCP/IP networking. It is the result of more than 10 years of teaching a university capstone course in hands-on cyber security.

It begins by showing how to build a range of Windows and Linux workstations, including CentOS, Mint, OpenSuSE, and Ubuntu systems. These can be physical or virtual systems built with VMWare Workstation or VirtualBox. Kali Linux is introduced and Metasploit is used to attack these systems, including EternalBlue and attacks against Internet Explorer, Firefox, Java, and Adobe Flash Player. These attacks all leave traces on the target and the network that can be found by a savvy defender, and these methods are demonstrated.

This interplay between setup, attack, and defense forms the core of the book. It continues through the process of setting up realistic networks with DNS servers and Windows Active Directory. Windows systems can be managed remotely using SMB, RPC, and WinRM; WMI is introduced, including the use of WMI to monitor systems. The Windows domain is then attacked, and techniques to escalate privileges from local user to domain user to domain administrator are developed. Tools like Mimikatz, Responder, and John the Ripper are used to obtain credentials, and hashes are passed across the domain. Linux systems are attacked next, and Dirty COW is demonstrated. To detect these attacks, a defender can turn to system logs; the reader will learn how logs are stored on Windows and Linux and how they can be made to interoperate. Sysmon is introduced and PowerShell used to query these logs.

An attacker with access to a system generally wants to maintain access to that system; this can be done using malware. Common vectors for persistence are demonstrated, including the registry, WMI persistence, and Kerberos golden tickets. A defender aware of these techniques can block or detect these attacks. An administrator can use PowerShell to search the domain to detect persistence mechanisms, firewall rules can be deployed to reduce lateral movement, and LAPS can be deployed to protect local accounts.

Of course, networks are built to provide services to users, so the book continues with an introduction to common services, including SSH, FTP, Windows file sharing, and Remote Desktop. Next are web servers, both IIS and Apache. These are configured, including using signed SSL/TLS certificates, attacked via a range of techniques, and defended with tools like ModSecurity. Real networks do not use a flat network topology, so network firewalls based on IPFire are introduced to separate the network into components and filter traffic in and out of the network. Databases are included in the network, and intrusion detection systems used to defend the network. The book concludes with an introduction to PHP and PHP-based web applications including WordPress, Joomla! and phpMyAdmin.

About the Systems

The book covers systems as they were used between 2011 and 2017. These systems should be patched now, so showing how to attack them today poses little risk to currently deployed systems. Back in the day, though, these systems were vulnerable to these exploits even though they were fully patched at the time. The defensive techniques discussed throughout the book retain their value and can be used to defend current systems even from new attacks.

About the Book

This book is designed for readers who are comfortable with Windows, Linux, and networking who want to learn the operational side of cyber security. It is meant to be read hand in hand with systems; indeed, the only way to learn cyber operations is to lay hands on a keyboard and work. Set up the various systems described in the book, try out the attacks, and look for the traces left by the attacks. Initially you may want to follow the text closely; but as you gain proficiency, it is better to use the text only as a guide and starting place for your own explorations.

I have taught a university capstone course in cyber security since 2004, and this book evolved from that course. It provides the reader a comprehensive introduction to hands-on cyber operations. It contains more material than can be comfortably covered in a semester, and yet, despite its size, it is far from exhaustive.

The book includes online supplementary material at https://www.apress.com/us/book/9781484242933. There you can find additional notes for each chapter, along with exercises that can be used either by an intrepid individual reader or by someone teaching a course.

Formatting

One problem with writing a book that includes computer output is that sometimes the screen output is wider than the page. Wherever possible, the text reproduces exactly what appears as the output from a command. However, when the output of a line is longer than the line on a page, I have taken the liberty of editing and formatting the result to make it easier for the reader. As an example, the raw output might look like the following.

```
msf exploit(ms17_010_eternalblue) > show payloads

Compatible Payloads
===================

 Name                                Disclosure Date  Rank    Description
 ----                                ---------------  ----    -----------
 generic/custom                                       normal  Custom Payload
 generic/shell_bind_tcp                               normal  Generic
 Command Shell, Bind TCP Inline
 generic/shell_reverse_tcp                            normal  Generic
 Command Shell, Reverse TCP Inline
 windows/x64/exec                                     normal  Windows x64
 Execute Command
 windows/x64/loadlibrary                              normal  Windows x64
 LoadLibrary Path

...Output Deleted ...

 windows/x64/meterpreter/reverse_http                 normal  Windows
 Meterpreter (Reflective Injection x64), Windows x64 Reverse HTTP Stager
 (wininet)
```

```
 windows/x64/meterpreter/reverse_https                  normal  Windows
 Meterpreter (Reflective Injection x64), Windows x64 Reverse HTTP Stager
 (wininet)
 windows/x64/meterpreter/reverse_tcp                    normal  Windows
 Meterpreter (Reflective Injection x64), Windows x64 Reverse TCP Stager

...Output Deleted ...
```

Instead, the text reads like the following.

```
msf exploit(ms17_010_eternalblue) > show payloads

Compatible Payloads
===================

 Name                          Rank    Description
 ----                          ----    -----------
 generic/custom                normal  Custom Payload
 generic/shell_bind_tcp        normal  Generic Command Shell, Bind TCP
                                       Inline
 generic/shell_reverse_tcp     normal  Generic Command Shell, Reverse TCP
                                       Inline
 windows/x64/exec              normal  Windows x64 Execute Command
 windows/x64/loadlibrary       normal  Windows x64 LoadLibrary Path

...Output Deleted ...

 windows/x64/meterpreter       normal  Windows Meterpreter
  /reverse_http                         (Reflective Injection x64), Windows
                                        x64 Reverse HTTP Stager (wininet)
 windows/x64/meterpreter       normal  Windows Meterpreter (Reflective
  /reverse_https                        Injection x64), Windows x64 Reverse
                                        HTTP Stager (wininet)
 windows/x64/meterpreter       normal  Windows Meterpreter (Reflective
  /reverse_tcp                          Injection x64), Windows x64 Reverse
                                           TCP Stager

... Output Deleted ...
```

I hope you agree that the latter is more readable.

Contacting the Author

You can reach Mike O'Leary at moleary@towson.edu. If you are a student or a faculty member participating at a Collegiate Cyber Defense exercise and you find this book helpful, I would love to hear from you.

CHAPTER 1

System Setup

Introduction

Cyber operations is about the configuration, defense, and attack of real systems. This text focuses on systems that were deployed between 2011 and 2017.

To configure, attack, and defend systems, a testing laboratory is required. Such a laboratory must not only allow systems to be built and run but must provide a way to segregate them from the wider Internet; after all, older systems are known to be vulnerable to public exploits. One excellent solution is virtualization. A range of virtualization solutions exists; two commonly deployed solutions are VMWare and VirtualBox. This chapter begins with an introduction to these virtualization solutions.

The chapter describes the major Windows desktop and server operating systems released between 2011 and 2017; it also includes major releases from the CentOS, OpenSuSE, Ubuntu, and Mint Linux distributions. The Notes and References section provides download locations for the various Linux distributions. This chapter shows how to build virtual machines running these operating systems.

A functioning computer system is more than just its operating system though; its entire ecosystem of installed applications must be considered. Desktop systems generally include a browser as well as plugins for various kinds of active web content. This chapter shows how to install three commonly used programs: Firefox, Java, and Adobe Flash Player on Windows and Linux workstations. The Notes and References lists download locations for these tools.

One advantage of modern operating systems and many major software packages is that they automatically download and install the latest security patches, often without user interaction. In almost every circumstance, this is a good thing. To keep test systems at a preferred patch level, this functionality must be controlled or disabled.

When this chapter is complete, the reader will have set up and configured a fully functional testing laboratory that can be used to run Windows and Linux virtual machines as they were deployed on a selected date between 2011 and 2017.

Virtualization Tools

A good testing laboratory needs a wide range of systems. Rather than use dedicated hardware for each system, it is much simpler to build systems using virtualization. Two of the most common tools for operating system virtualization are VMWare Workstation and VirtualBox, while other

M. O'Leary, *Cyber Operations*, https://doi.org/10.1007/978-1-4842-4294-0_1

choices include ProxMox, Hyper-V, Parallels, QEMU, and Xen. This book focuses solely on the first two of these. VMWare Workstation is a long-standing, solid commercial product that runs on Windows and Linux; it has a free version called VMWare Player with reduced functionality. VirtualBox is a free, open source alternative; it runs on Windows Linux, Macintosh, and Solaris. In its current version, it is comparable to VMWare Workstation in functionality.

VMWare Workstation

The simplest way to learn about VMWare Workstation is to dive right in by installing and running a guest operating system.

Installing a VMWare Guest

Grab the install disc for a Linux distribution or a Windows system, and save that `.iso` file in some convenient location.[1] Launch VMWare Workstation. If the home tab appears, select "Create a New Virtual Machine"; if it does not, then the same option is available from the File menu.

VMWare Workstation begins the process of creating a new virtual machine by presenting the user with the "New Virtual Machine Wizard." The "Typical" configuration is usually sufficient. The first question is the location of the install media; provide the location of the saved `.iso` file for the "Installer disc image file (iso)." In most, though not all cases, VMWare Workstation recognizes the operating system on the disc image. When VMWare Workstation moves to install a recognized operating system, it uses "Easy Install" and makes several choices for the user. This automated process is often convenient, however it precludes the user from choosing some things, like the system partition table or the precise collection of installed software; this can occasionally cause difficulty later.

When VMWare Workstation is installing a Windows system, it provides a dialog box that allows the user to enter the Windows product key, the precise version of Windows (e.g., Windows 8 Professional), and the new system's user and password. When a Linux system like CentOS is being installed, VMWare instead asks for information about a system user: the user's full name, the username, and the password for that user. The same password for the user is also used for the root account on the system. In either case, VMWare Workstation then asks for both the name of the virtual machine and the location in which it will be stored. The VMWare Workstation name is separate and distinct from any host name of the system; it is used solely by VMWare Workstation to generate the names of the files that comprise the virtual machine and will appear as the machine's title within VMWare Workstation. When selecting the location of those files, note that there are many files for each virtual machine, so it is a very good idea to store each system in its own separate directory.

[1]The Notes and References section at the end of the chapter provides links to locations containing installation discs for CentOS, Kali, Mint, Ubuntu, and OpenSuSE as well as to evaluation copies of Microsoft operating systems.