

ATTACKING NETWORK PROTOCOLS

**A Hacker's Guide to
Capture, Analysis,
and Exploitation**

by James Forshaw



**no starch
press**

San Francisco

ATTACKING NETWORK PROTOCOLS. Copyright © 2018 by James Forshaw.

All rights reserved. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage or retrieval system, without the prior written permission of the copyright owner and the publisher.

ISBN-10: 1-59327-750-4

ISBN-13: 978-1-59327-750-5

Publisher: William Pollock

Production Editor: Laurel Chun

Cover Illustration: Garry Booth

Interior Design: Octopod Studios

Developmental Editors: Liz Chadwick and William Pollock

Technical Reviewers: Cliff Janzen

Additional Technical Reviewers: Arrigo Triulzi and Peter Gutmann

Copyeditor: Anne Marie Walker

Compositors: Laurel Chun and Meg Sneeringer

Proofreader: Paula L. Fleming

Indexer: BIM Creatives, LLC

For information on distribution, translations, or bulk sales, please contact No Starch Press, Inc. directly:

No Starch Press, Inc.

245 8th Street, San Francisco, CA 94103

phone: 1.415.863.9900; info@nostarch.com

www.nostarch.com

Library of Congress Control Number: 2017954429

No Starch Press and the No Starch Press logo are registered trademarks of No Starch Press, Inc. Other product and company names mentioned herein may be the trademarks of their respective owners. Rather than use a trademark symbol with every occurrence of a trademarked name, we are using the names only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The information in this book is distributed on an “As Is” basis, without warranty. While every precaution has been taken in the preparation of this work, neither the author nor No Starch Press, Inc. shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in it.

About the Author

James Forshaw is a renowned computer security researcher at Google Project Zero, with more than ten years of experience in analyzing and exploiting application network protocols. His skills range from cracking game consoles to exposing complex design issues in operating systems, especially Microsoft Windows, which earned him the top bug bounty of \$100,000 and placed him as the #1 researcher on Microsoft Security Response Center's (MSRC) published list. He's the creator of the network protocol analysis tool, Canape, which was developed from his years of experience. He's been invited to present his novel security research at global security conferences such as BlackHat, CanSecWest and Chaos Computer Congress.

About the Technical Reviewer

Since the early days of Commodore PET and VIC-20, technology has been a constant companion (and sometimes an obsession!) to Cliff Janzen. Cliff discovered his career passion when he moved to information security in 2008 after a decade of IT operations. Since then, Cliff has had the great fortune to work with and learn from some of the best people in the industry, including Mr. Forshaw and the fine people at No Starch during the production of this book. He is happily employed as a security consultant, doing everything from policy review to penetration tests. He feels lucky to have a career that is also his favorite hobby and a wife who supports him.

This page intentionally left blank

BRIEF CONTENTS

Foreword by Katie Moussouris	xv
Acknowledgments	xvii
Introduction	xix
Chapter 1: The Basics of Networking	1
Chapter 2: Capturing Application Traffic	11
Chapter 3: Network Protocol Structures	37
Chapter 4: Advanced Application Traffic Capture	63
Chapter 5: Analysis from the Wire	79
Chapter 6: Application Reverse Engineering	111
Chapter 7: Network Protocol Security	145
Chapter 8: Implementing the Network Protocol	179
Chapter 9: The Root Causes of Vulnerabilities	207
Chapter 10: Finding and Exploiting Security Vulnerabilities	233
Appendix: Network Protocol Analysis Toolkit	277
Index	293

This page intentionally left blank

CONTENTS IN DETAIL

FOREWORD by Katie Moussouris	xv
-------------------------------------	-----------

ACKNOWLEDGMENTS	xvii
------------------------	-------------

INTRODUCTION	xix
---------------------	------------

Why Read This Book?	xx
What's in This Book?	xx
How to Use This Book.	xxii
Contact Me	xxii

1	
THE BASICS OF NETWORKING	1

Network Architecture and Protocols	1
The Internet Protocol Suite	2
Data Encapsulation	4
Headers, Footers, and Addresses	4
Data Transmission	6
Network Routing	7
My Model for Network Protocol Analysis	8
Final Words	10

2	
CAPTURING APPLICATION TRAFFIC	11

Passive Network Traffic Capture	12
Quick Primer for Wireshark.	12
Alternative Passive Capture Techniques.	14
System Call Tracing.	14
The strace Utility on Linux.	16
Monitoring Network Connections with DTrace	16
Process Monitor on Windows	18
Advantages and Disadvantages of Passive Capture	19
Active Network Traffic Capture	20
Network Proxies	20
Port-Forwarding Proxy	21
SOCKS Proxy	24
HTTP Proxies.	29
Forwarding an HTTP Proxy	29
Reverse HTTP Proxy	32
Final Words	35

3

NETWORK PROTOCOL STRUCTURES

37

Binary Protocol Structures	38
Numeric Data	38
Booleans	41
Bit Flags	41
Binary Endian	41
Text and Human-Readable Data	42
Variable Binary Length Data	47
Dates and Times	49
POSIX/Unix Time	50
Windows FILETIME	50
Tag, Length, Value Pattern	50
Multiplexing and Fragmentation	51
Network Address Information	52
Structured Binary Formats	53
Text Protocol Structures	54
Numeric Data	55
Text Booleans	55
Dates and Times	55
Variable-Length Data	56
Structured Text Formats	56
Encoding Binary Data	59
Hex Encoding	59
Base64	60
Final Words	62

4

ADVANCED APPLICATION TRAFFIC CAPTURE

63

Rerouting Traffic	64
Using Traceroute	64
Routing Tables	65
Configuring a Router	66
Enabling Routing on Windows	67
Enabling Routing on *nix	67
Network Address Translation	68
Enabling SNAT	68
Configuring SNAT on Linux	69
Enabling DNAT	70
Forwarding Traffic to a Gateway	71
DHCP Spoofing	71
ARP Poisoning	74
Final Words	77

5

ANALYSIS FROM THE WIRE

79

The Traffic-Producing Application: SuperFunkyChat	80
Starting the Server	80
Starting Clients	80
Communicating Between Clients	81

A Crash Course in Analysis with Wireshark	81
Generating Network Traffic and Capturing Packets	83
Basic Analysis	84
Reading the Contents of a TCP Session	85
Identifying Packet Structure with Hex Dump	86
Viewing Individual Packets	87
Determining the Protocol Structure	88
Testing Our Assumptions	89
Dissecting the Protocol with Python	90
Developing Wireshark Dissectors in Lua	95
Creating the Dissector	98
The Lua Dissection	99
Parsing a Message Packet	100
Using a Proxy to Actively Analyze Traffic	103
Setting Up the Proxy	103
Protocol Analysis Using a Proxy	105
Adding Basic Protocol Parsing	107
Changing Protocol Behavior	108
Final Words	110

6 **APPLICATION REVERSE ENGINEERING** 111

Compilers, Interpreters, and Assemblers	112
Interpreted Languages	112
Compiled Languages	113
Static vs. Dynamic Linking	113
The x86 Architecture	114
The Instruction Set Architecture	114
CPU Registers	116
Program Flow	118
Operating System Basics	119
Executable File Formats	119
Sections	120
Processes and Threads	120
Operating System Networking Interface	121
Application Binary Interface	123
Static Reverse Engineering	125
A Quick Guide to Using IDA Pro Free Edition	125
Analyzing Stack Variables and Arguments	128
Identifying Key Functionality	129
Dynamic Reverse Engineering	134
Setting Breakpoints	135
Debugger Windows	135
Where to Set Breakpoints?	137
Reverse Engineering Managed Languages	137
.NET Applications	137
Using ILSpy	138
Java Applications	141
Dealing with Obfuscation	143
Reverse Engineering Resources	144
Final Words	144

7

NETWORK PROTOCOL SECURITY

145

Encryption Algorithms	146
Substitution Ciphers	147
XOR Encryption	148
Random Number Generators	149
Symmetric Key Cryptography	149
Block Ciphers	150
Block Cipher Modes	152
Block Cipher Padding	155
Padding Oracle Attack	156
Stream Ciphers	158
Asymmetric Key Cryptography	159
RSA Algorithm	160
RSA Padding	162
Diffie–Hellman Key Exchange	162
Signature Algorithms	164
Cryptographic Hashing Algorithms	164
Asymmetric Signature Algorithms	165
Message Authentication Codes	166
Public Key Infrastructure	169
X.509 Certificates	169
Verifying a Certificate Chain	170
Case Study: Transport Layer Security	172
The TLS Handshake	172
Initial Negotiation	173
Endpoint Authentication	174
Establishing Encryption	175
Meeting Security Requirements	176
Final Words	178

8

IMPLEMENTING THE NETWORK PROTOCOL

179

Replaying Existing Captured Network Traffic	180
Capturing Traffic with Netcat	180
Using Python to Resend Captured UDP Traffic	182
Repurposing Our Analysis Proxy	183
Repurposing Existing Executable Code	188
Repurposing Code in .NET Applications	189
Repurposing Code in Java Applications	193
Unmanaged Executables	195
Encryption and Dealing with TLS	200
Learning About the Encryption In Use	200
Decrypting the TLS Traffic	201
Final Words	206

9

THE ROOT CAUSES OF VULNERABILITIES

207

Vulnerability Classes	208
Remote Code Execution	208
Denial-of-Service	208
Information Disclosure	209
Authentication Bypass	209
Authorization Bypass	209
Memory Corruption Vulnerabilities	210
Memory-Safe vs. Memory-Unsafe Programming Languages	210
Memory Buffer Overflows	210
Out-of-Bounds Buffer Indexing	216
Data Expansion Attack	217
Dynamic Memory Allocation Failures	217
Default or Hardcoded Credentials	218
User Enumeration	218
Incorrect Resource Access	219
Canonicalization	220
Verbose Errors	221
Memory Exhaustion Attacks	222
Storage Exhaustion Attacks	223
CPU Exhaustion Attacks	224
Algorithmic Complexity	224
Configurable Cryptography	226
Format String Vulnerabilities	227
Command Injection	228
SQL Injection	228
Text-Encoding Character Replacement	229
Final Words	231

10

FINDING AND EXPLOITING SECURITY VULNERABILITIES

233

Fuzz Testing	234
The Simplest Fuzz Test	234
Mutation Fuzzer	235
Generating Test Cases	235
Vulnerability Triaging	236
Debugging Applications	236
Improving Your Chances of Finding the Root Cause of a Crash	243
Exploiting Common Vulnerabilities	245
Exploiting Memory Corruption Vulnerabilities	246
Arbitrary Memory Write Vulnerability	253
Writing Shell Code	255
Getting Started	256
Simple Debugging Technique	258
Calling System Calls	259

Executing the Other Programs	263
Generating Shell Code with Metasploit	265
Memory Corruption Exploit Mitigations	266
Data Execution Prevention	267
Return-Oriented Programming Counter-Exploit	268
Address Space Layout Randomization (ASLR)	270
Detecting Stack Overflows with Memory Canaries	273
Final Words	276

NETWORK PROTOCOL ANALYSIS TOOLKIT 277

Passive Network Protocol Capture and Analysis Tools	278
Microsoft Message Analyzer	278
TCPDump and LibPCAP	278
Wireshark	279
Active Network Capture and Analysis	280
Canape	280
Canape Core	281
Mallory	281
Network Connectivity and Protocol Testing	282
Hping	282
Netcat	282
Nmap	282
Web Application Testing	283
Burp Suite	283
Zed Attack Proxy (ZAP)	284
Mitmproxy	284
Fuzzing, Packet Generation, and Vulnerability Exploitation Frameworks	285
American Fuzzy Lop (AFL)	285
Kali Linux	286
Metasploit Framework	286
Scapy	287
Sulley	287
Network Spoofing and Redirection	287
DNSMasq	287
Ettercap	287
Executable Reverse Engineering	288
Java Decompiler (JD)	288
IDA Pro	289
Hopper	289
ILSpy	290
.NET Reflector	290

INDEX 293