Practical Video Game Bots

Automating Game Processes using C++, Python, and AutoIt

Ilya Shpigor

Apress®

Practical Video Game Bots: Automating Game Processes using C++, Python, and AutoIt

Ilya Shpigor
St. Petersburg, c.St-Peterburg, Russia

ISBN-13 (pbk): 978-1-4842-3735-9 ISBN-13 (electronic): 978-1-4842-3736-6
https://doi.org/10.1007/978-1-4842-3736-6

Library of Congress Control Number: 2018954729

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Steve Anglin
Development Editor: Matthew Moodie
Coordinating Editor: Mark Powers

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 editorial@apress.com; for reprint, paperback, or audio rights, please email bookpermissions@springernature.com.

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/9781484237359. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

Table of Contents

About the Author vii

About the Technical Reviewer ix

Acknowledgments xi

Preface xiii

Introduction xv

Chapter 1: Overview of Bots 1

Purpose of Bots 1

Game Application 3

Types of Bots 7

Community Classification 7

Developer Classification 9

Bot Comparison 12

Summary 15

Chapter 2: Clicker Bots 17

Developer Tools 17

Programming Language 18

Image Processing Libraries 18

Image Analysis Tool 19

Source Code Editors 19

API Hooking 19

OS-Level Data Embedding 20

Keystroke Simulation 23

Mouse Simulation 31

OS-Level Data Embedding Summary 36

Output Device Capture 36
Windows Graphics Device Interface 36
AutoIt Analysis Functions 38
Advanced Image Analysis Libraries 46
Capturing Output Device Summary 56
Example with Lineage 2 56
Lineage 2 Overview 56
Bot Implementation 58
Lineage 2 Summary 69
Protection Approaches 70
Test Application 71
Analysis of Actions 72
Process Scanner 81
Keyboard State Check 89
Protection Summary 94

Chapter 3: In-game Bots 95

Tools 95
Programming Language 95
Debugger 96
Memory Analyzing Tools 97
Process Memory Analysis 97
Process Memory Overview 97
Variable Searching 105
Process Memory Analysis Summary 117
Process Memory Access 117
Open Process 117
Read and Write Operations 121
TEB and PEB Access 124
Heap Access 139
Process Memory Access Summary 142

Example with Diablo 2 142
Bot Overview 145
Diablo 2 Memory Analysis 146
Bot Implementation 160
Further Improvements 167
Example Summary 170
Protection Approaches 171
Test Application 171
Approaches Against Analysis 178
Approaches Against Bots 197
Protection Approaches Summary 207

Chapter 4: Out-game Bots 209

Tools 209
Programming Language 209
Network Analyzer 211
Windows Configuration 211
Internet Protocols 213
Communication Tasks 213
TCP/IP Stack 217
Packet Analysis 221
Test Application 221
Packet Capture 226
UDP Connection 233
Example with NetChess 235
Bot Overview 239
NetChess Traffic Analysis 239
Bot Implementation 247
Assessing the Bot 251
Protection Approaches 252
Cryptographic System 252
Test Application 253

XOR Cipher 254
Triple DES Cipher 259
AES Cipher 264
RSA Cipher 267
Detecting Out-game Bots 273

Chapter 5: Extra Techniques 275

Input Device Emulation 275
Input Device Emulation Tools 275
Keyboard Emulation 276
Keyboard Modifiers 282
Mouse Emulation 285
Keyboard and Mouse Emulation 290
Input Device Emulation Summary 296
OS-Level Interception Data 297
OS-Level Interception Data Tools 297
Test Application 298
DLL Import 299
API Hooking Techniques 302
Proxy DLL 302
Example of Proxy DLL 305
API Patching 309
Example of API Patching 311
OS-Level Interception Data Summary 317

Index 319

About the Author

Ilya Shpigor is a software developer and open source enthusiast. He has significant experience in such domains as Embedded Systems, Information Security, and Real-Time Computing.

Ilya currently works in the automotive industry. He develops security systems for Ethernet networks in cars. Before that, he developed intrusion detection systems, flight simulators, and control systems for sea ships. Also, he has participated in the Wine open source project and ALT Linux distribution.

Ilya is interested in automating routine tasks and researching the capacities of different programming languages to solve specific problems. In his free time, he explores software vulnerabilities and AI approaches.

About the Technical Reviewer

Massimo Nardone has more than 24 years of experience in Security, Web/Mobile Development, Cloud, and IT Architecture. His true IT passions are Security and Android.

He has been programming and teaching how to program with Android, Perl, PHP, Java, VB, Python, C/C++, and MySQL for more than 20 years.

He holds a Master of Science degree in Computing Science from the University of Salerno, Italy.

He has worked as a Project Manager, Software Engineer, Research Engineer, Chief Security Architect, Information Security Manager, PCI/SCADA Auditor, and Senior Lead IT Security/Cloud/SCADA Architect for many years.

His technical skills include Security, Android, Cloud, Java, MySQL, Drupal, Cobol, Perl, Web and Mobile development, MongoDB, D3, Joomla, Couchbase, C/C++, WebGL, Python, Pro Rails, Django CMS, Jekyll, Scratch, etc.

He has worked as a visiting lecturer and supervisor for exercises at the Networking Laboratory of the Helsinki University of Technology (Aalto University). He holds four international patents (PKI, SIP, SAML, and Proxy areas).

He currently works as Chief Information Security Officer (CISO) for Cargotec Oyj and is a member of the ISACA Finland Chapter Board.

Massimo has reviewed more than 45 IT books for different publishing companies and is the coauthor of *Pro Android Games* (Apress, 2015), *Pro JPA 2 in Java EE 8* (Apress, 2018), and *Beginning EJB in Java EE 8* (Apress, 2018).

Acknowledgments

A special thank you to Svetlana Zalogina, who reviewed the first chapters of this book and provided many style recommendations. Also, I would like to thank Danila Bogdanov and Emil Shaykhilislamov, who pointed out my mistakes and gave me advice on how to cover the game bot topic better. Thanks to Ruslan Piasetskyi, who explained to me some subtleties of the cryptography domain.

Preface

This is not a guide on how to cheat and violate rules in video games. This is a book about approaches to automating a game process and protecting it against automation.

We will consider applications that play video games in your place; they are named bots. You will find here a classification of such applications by their internal mechanics. The book covers most methods and technologies that are used by bot developers. Also, the various approaches of anticheating systems are considered here.

This book provides solutions and useful advices for such topics as process automation, reverse engineering, encryption, and network applications. Modern bots use technologies in all these domains.

Introduction

Sometimes when you play your favorite video game, you can find yourself repeating simple actions. Perhaps this process reminds you of working with old manual machines. You would mount a piece of metal, press the button to launch the drill, pull the lever down, and so forth. But wait a minute. We live in the 21st century, and long before us people have learned ways to automate simple, monotonous actions. These thoughts occurred to me while I was playing my favorite video game.

After that, I decided to start looking for ways to automate my game process. I have visited plenty of forums and websites. Most of the applications for game automation that I found contained malicious software. Some of them were virus-free, but they did not work at all. During my searches, people with strange nicknames suggested that I buy these black magic applications that should solve all my problems. But it seems pretty weird to buy something from an anonymous person over the Internet without any guarantees. Further, I realized why bot developers prefer to hide their names. Thus, my searches failed.

My next step was an attempt to implement a bot myself. But I faced a shortage of systematic documentation about the topic, despite the fact that bot applications often solve difficult algorithmic tasks and are based on several information technology domains. The situation looked very strange, because this kind of software can be very complex, and moreover, bot development has a long history. Enthusiasts and professional software developers found a lot of solutions and approaches to effectively solving this task. Why didn't anybody care about sharing this kind of information?

This book is an attempt to overcome this information vacuum around the topic of bot development. You will find a bot classification here that I developed from my experience and research. We will consider the internal mechanisms of different kinds of bots and will try to write simple prototypes. You will learn about tools for bot development as well as anticheating systems for preventing usage of the bots.

The book will be interesting to all players who want to discover a new sense and approach to the game process. It will also be useful for players who do not care about bot application internals but just want to buy one and use it. You will learn about the available kinds of bots and which exploitation issues you may face. I hope everybody will find something interesting and new in this book.