CODELIVLY MALWARE SERIES · VOLUME 2

Malware Analysis for Beginners

Practical Static & Dynamic Malware Analysis

BUILD A LAB · ANALYZE SAFELY · EXTRACT IOCS · WRITE THE REPORT

STATIC · DYNAMIC · BEHAVIORAL · REPORTING

Codelivly www.codelivly.com

LEARN. PRACTICE. HACK. DEFEND.

Malware Analysis for Beginners

Volume 2 — Practical Static & Dynamic Malware Analysis
Codelivly Malware Series

Copyright © 2026 Codelivly. All Rights Reserved.

No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other non-commercial uses permitted by copyright law.

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 Codelivly 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.

Trademarks: All product names, tool names, logos, and brands referenced in this book are the property of their respective owners and are used for identification and educational purposes only. MITRE ATT&CK® is a registered trademark of The MITRE Corporation.

Published by Codelivly

Publisher: Codelivly
Website: www.codelivly.com
Series: Codelivly Malware Series
Volume: 2 — Practical Static & Dynamic Malware Analysis
Edition: First Edition, 2026
Tagline: Learn. Practice. Hack. Defend.

This volume builds on *Volume 1 — Understanding Modern Malware from a Defender's Perspective*. Reading Volume 1 first is helpful but not required; every concept used here is reintroduced from first principles.

© 2026 Codelivly. All Rights Reserved.

Legal & Ethical Disclaimer

This book is intended **solely for cybersecurity education, malware analysis, digital forensics, incident response, and defensive research**. Its purpose is to help students, analysts, and defenders learn how to safely examine suspicious software so they can detect, contain, and remediate real threats.

IMPORTANT

All practical exercises in this book must be performed **only inside isolated laboratory environments** using authorized samples or simulations. This book does **not** teach how to create, modify, weaponize, deploy, or distribute malware, and it deliberately omits offensive tooling and working malicious code.

By reading this book you acknowledge and agree that:

- You will analyze malware and suspicious samples **only within isolated, authorized lab environments** that you own or are explicitly permitted to use — never on production systems.
- You will comply with all applicable local, national, and international laws, and with all organizational policies that govern your work.
- You will never use knowledge from this book to gain unauthorized access to systems, to damage data, or to harm any person or organization.
- Live malware is dangerous. Mishandling it can destroy data and spread infections. You are solely responsible for handling samples safely.

Neither Codelivly nor the authors endorse or encourage unauthorized access, malware deployment, or any other malicious activity. If you are unsure whether an activity is legal or authorized, stop and seek guidance before proceeding.

© 2026 Codelivly. All Rights Reserved.

Foreword

The first time most people open a real malware sample, they feel two things at once: fascination and fear. Fascination, because a few kilobytes of code can do so much. Fear, because it is genuinely dangerous and it is easy to make a mistake. This book was written to replace that fear with *method*.

Malware analysis is not magic and it is not reserved for a gifted few. It is a craft, and like any craft it is built from repeatable steps, the right tools, and disciplined habits. The difference between a beginner who feels lost and an analyst who feels in control is almost never raw talent — it is process. Give a nervous beginner a safe lab, a checklist, and a clear order of operations, and within weeks they are pulling indicators out of samples that once looked like an impenetrable wall of hex.

What you hold is a hands-on apprenticeship in that process. It takes you from an empty computer to a fully isolated analysis lab, then walks you through static analysis, dynamic analysis, behavioral analysis, indicator extraction, and professional reporting — the exact workflow working analysts use every day. Every technique is tied back to the only thing that matters in the end: helping defenders find, understand, and stop threats.

— The Codelivly Editorial Team

© 2026 Codelivly. All Rights Reserved.

Preface

This is **Volume 2** of the Codelivly Malware Series. Where Volume 1 explained *what malware is*, this volume teaches *how to analyze it* — safely, systematically, and to a professional standard. It assumes no reverse-engineering background. If you can install software and follow steps carefully, you are ready.

HOW THIS BOOK IS ORGANIZED

The book has six parts. **Part I** introduces malware analysis and walks you through building a safe lab. **Part II** gives you the Windows internals and PE-format knowledge that all analysis rests on. **Part III** teaches static analysis — examining a file without running it. **Part IV** teaches dynamic analysis — safely running a sample and watching what it does. **Part V** turns raw observations into behavior, indicators, and a finished report. **Part VI** applies the whole workflow to famous real-world malware from a defender's point of view.

HOW TO USE THIS BOOK

Read the parts in order the first time through; each builds on the last. Every chapter has learning objectives, step-by-step tutorials, diagrams, tables, callout boxes, an ATT&CK mapping where relevant, a full hands-on lab, review questions, and a practical assignment. **Do the labs.** Reading about analysis makes you informed; performing it makes you an analyst.

> **BEFORE YOU BEGIN**
>
> Do not download or execute any real malware until you have built and verified the isolated lab in Chapter 2. Until then, use only the benign demonstration binaries and simulations referenced in the exercises.

A NOTE ON CONVENTIONS

Commands and file paths appear in `monospace`. Where a behavior maps to the MITRE ATT&CK® framework, the technique ID is shown like this: **T1547 Boot/Logon Autostart**. Screenshot placeholders mark where you should capture your own screen while doing a lab — building that habit is part of learning to document your work.

© 2026 Codelivly. All Rights Reserved.

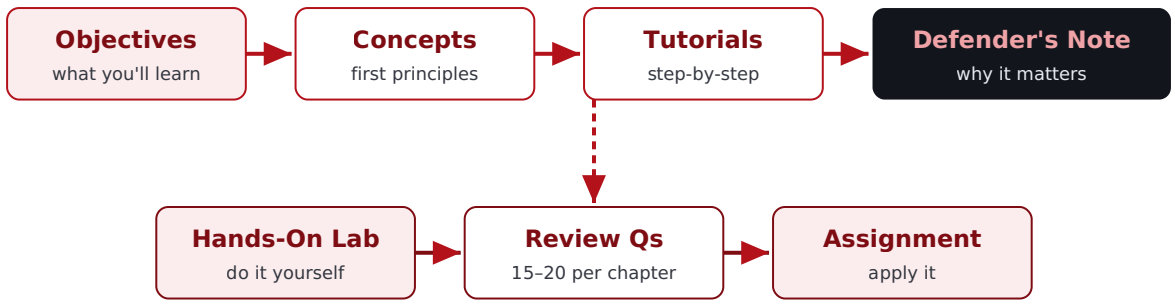
Table of Contents

Part I — Introduction 11
1 What Is Malware Analysis? 12
2 Building a Safe Malware Lab 24
Part II — Windows Internals 37
3 Windows Architecture 38
4 The Portable Executable (PE) Format 50
Part III — Static Malware Analysis 62
5 Initial File Triage 63
6 Strings Analysis 73
7 PEStudio 82
8 Detect It Easy (DIE) 90
Part IV — Dynamic Malware Analysis 129
9 Process Monitoring 130
10 Network Analysis 98
11 Registry Analysis 109
12 File System Monitoring 119
Part V — Behavioral Analysis & Reporting 141
13 Malware Behaviors 142
14 Indicators of Compromise (IOCs) 152
15 Malware Analysis Report Writing 161
Part VI — Case Studies 172
16 Defensive Case Studies (WannaCry · Emotet · TrickBot · Zeus · Mirai · NotPetya) 173
Appendices 186
A Malware Analysis Checklist 187
B IOC Collection Checklist 189
C Windows API Reference (Analysis Context) 190
D Common Registry Keys & PE Quick Reference 192
E Glossary, Tools, Roadmap & Further Reading 194
F Quick Reference Card 201

© 2026 Codelivly. All Rights Reserved.

How to Read This Book

Every chapter follows the same rhythm, so you always know where you are and what comes next.



Learn along the top row; prove it to yourself along the bottom.

Figure 0.1 — *The anatomy of every chapter in this book.*

THE THREE HABITS OF A GOOD ANALYST

Isolate — never analyze on a machine you care about. **Snapshot** — return to a known-clean state between runs. **Document** — if you didn't write it down, it didn't happen. These three habits appear in every chapter; make them reflex.

© 2026 Codelivly. All Rights Reserved.

Notation, Conventions & Icons

A quick key to the visual language used throughout the book, so every element is instantly readable.

Callout boxes

PRO TIP
A shortcut, best practice, or time-saver from the bench.

COMMON MISTAKE
A frequent beginner error and how to avoid it.

DEFENDER'S NOTE
How a concept connects to real detection and response.

IMPORTANT
A safety-critical point — read these carefully.

DID YOU KNOW?
Background or history that deepens understanding.

Other elements

Element	Meaning
`monospace`	Commands, file paths, code, and API names
T1547 Autostart	A MITRE ATT&CK® technique mapping
Screenshot placeholder box	Where you should capture your own screen during a lab
Figures	Diagrams that illustrate structure, flow, or behavior
Hands-On Lab	A full guided exercise with a 9-part structure

LAB ADDRESSES USED THROUGHOUT
Examples assume the lab from Chapter 2: a Windows victim at `10.0.0.10` and a Linux services VM at `10.0.0.20` running INetSim, on an isolated network with no route to the real internet.

© 2026 Codelivly. All Rights Reserved.

The Codelivly Malware Series

This book is one volume in a progression designed to take you from curious beginner to capable defender.

Volume	Focus	You come away able to…
Volume 1	Understanding Modern Malware	Explain what malware is, how it behaves, and how defenders think
Volume 2 (this book)	Practical Static & Dynamic Analysis	Build a lab and analyze samples end to end, producing IOCs and reports
Beyond	Reverse engineering & detection engineering	Read code, defeat obfuscation, and build advanced detections

Volume 2 stands on its own — every concept is introduced from first principles — but it assumes the mindset Volume 1 builds: that analysis exists to serve defense, and that safety and ethics are part of the craft, not constraints on it.

What you will build by the end

- A verified-isolated malware analysis lab you trust.
- A repeatable static-then-dynamic workflow you can run under pressure.
- A portfolio of complete, professional analysis reports on public research samples.
- The judgment to turn observations into durable, defensive detections.

HOW TO GET THE MOST FROM THIS BOOK

Read actively and *do the labs*. Reading about analysis makes you informed; performing it — safely, in your own lab — is what makes you an analyst. Every lab you finish is a step toward the portfolio that opens doors.

© 2026 Codelivly. All Rights Reserved.

PART I

Introduction

Before you touch a single sample, you need two things: a clear picture of what malware analysis actually is, and a lab where you can work without endangering anything you care about. Part I gives you both — the mindset and the machine.

© 2026 Codelivly. All Rights Reserved.

PART I · INTRODUCTION

01 What Is Malware Analysis?

Malware analysis is the disciplined process of examining a suspicious program to learn what it is, what it does, and how to detect and stop it — all without letting it harm anything real.

LEARNING OBJECTIVES

- Define malware analysis and explain the questions it exists to answer.
- Distinguish static from dynamic analysis, and basic from advanced analysis.
- Describe the end-to-end malware analysis workflow and where each technique fits.
- Identify the roles and career paths that use these skills.
- State the ethical and legal principles that govern safe analysis.

1.1 Goals of Malware Analysis

When a suspicious file lands on an analyst's desk, it arrives with questions attached. Is it actually malicious, or a false alarm? If it is malicious, what does it do — steal data, encrypt files, open a backdoor? How does it get in, and how does it survive a reboot? Who or what does it talk to? And most importantly for a defender: **how do we detect it everywhere else, and how do we get rid of it?** Malware analysis is simply the structured effort to answer those questions with evidence rather than guesswork.

It helps to separate the goals into three levels, because they call for different depth and different tools:

Goal	Question it answers	Typical output
Triage / verdict	Is this malicious, and how urgent?	A yes/no/maybe with confidence
Indicator extraction	What can we detect and block?	Hashes, domains, IPs, registry keys
Capability & intent	What does it actually do?	Behavior narrative, ATT&CK mapping
Attribution / context	Who is behind it, and is it targeted?	Threat-intel linkage (advanced)

© 2026 Codelivly. All Rights Reserved.

For a beginner, the first three goals are the entire job, and they are enough to make you genuinely useful to a security team. Attribution is a specialized, advanced pursuit that builds on the same foundations.

> **DEFENDER'S NOTE**
> Analysis is never done for its own sake. Every finding should map to a defensive action: a detection rule, a block, a hunt, or a remediation step. If a piece of analysis cannot change what a defender does, it can usually wait.

1.2 Static vs. Dynamic Analysis

The two fundamental approaches to analyzing a sample are **static analysis** and **dynamic analysis**. They are complementary — most real investigations use both, in that order.

Static analysis means examining the file *without executing it*. You look at its bytes, its structure, the human-readable strings inside it, the library functions it imports, and its metadata. Static analysis is safe (the code never runs), fast, and gives you a hypothesis. Its weakness is that modern malware hides itself with packing and obfuscation, so a static-only view is often incomplete.

Dynamic analysis means *running the sample in a safe, isolated environment* and watching what it does — the files it drops, the registry keys it writes, the processes it spawns, the network connections it makes. Dynamic analysis cuts straight through obfuscation, because the malware must reveal its true behavior to accomplish anything. Its weakness is risk (you are running live malware) and the fact that clever samples can detect the lab and refuse to run.

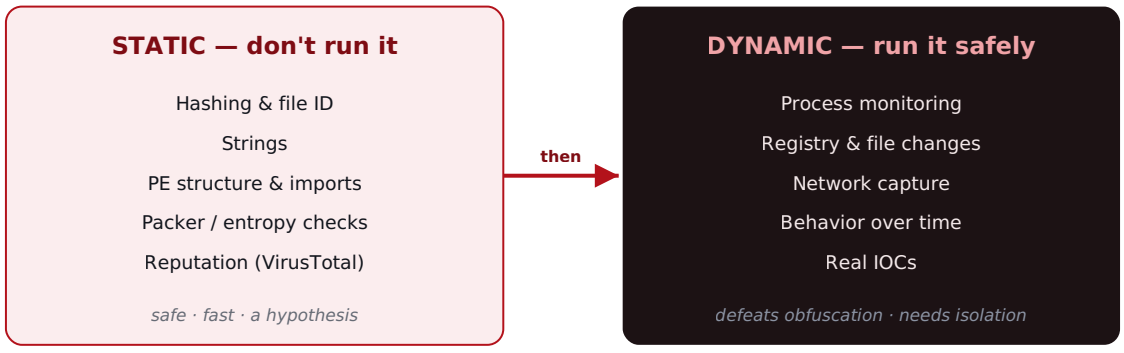


Figure 1.1 — *Static and dynamic analysis are partners: static forms the hypothesis, dynamic confirms it.*

© 2026 Codelivly. All Rights Reserved.

1.3 Basic vs. Advanced Analysis

Cutting the field a second way, each approach has a *basic* and an *advanced* tier. Basic analysis uses accessible tools and observation; advanced analysis uses a disassembler or debugger to read the code itself. This book focuses on the basic tiers, which cover the overwhelming majority of day-to-day work, and introduces advanced concepts so you know where to go next.

	Basic	Advanced
Static	Hashes, strings, PE headers, imports, packer detection	Disassembly in a tool like a decompiler; reading the code without running it
Dynamic	Process/registry/file/network monitoring in a sandbox	Debugging: single-stepping the code, setting breakpoints, unpacking by hand

PRO TIP

Beginners often rush toward disassembly because it looks impressive. Resist. Basic static and dynamic analysis answer most real questions faster, and they build the intuition that makes advanced analysis productive later.

1.4 The Malware Analysis Workflow

Professional analysis follows a repeatable pipeline. Each stage narrows the unknowns and feeds the next, and the whole thing is bracketed by safety: you start by getting the sample in safely and end by wiping the lab clean.

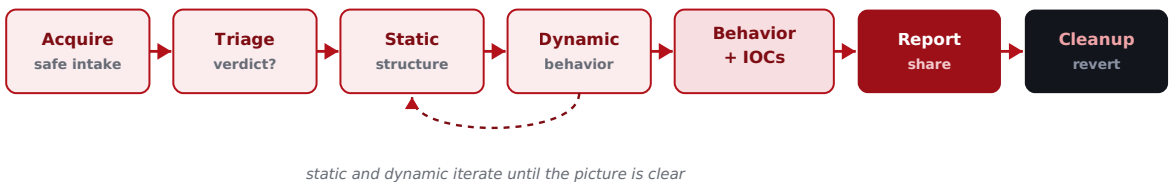


Figure 1.2 — *The malware analysis workflow followed throughout this book.*

You will recognize this pipeline as the skeleton of the book: Part III is "Static," Part IV is "Dynamic," and Part V is "Behavior + IOCs + Report." Chapter 2 handles "Acquire" and "Cleanup" by building the lab that makes both safe.

© 2026 Codelivly. All Rights Reserved.

1.5 Malware Analysis Roles

These skills are used under many job titles. Knowing the landscape helps you see where analysis fits and where you might specialize.

Role	How they use analysis
SOC Analyst (Tier 1–2)	Triage suspicious files and alerts; extract quick IOCs to contain incidents
DFIR Analyst	Analyze samples found during incident response; reconstruct what happened
Threat Hunter	Turn analysis findings into hunts across the fleet
Detection Engineer	Convert behaviors into durable detection rules (YARA, Sigma, EDR)
Malware Researcher / Reverse Engineer	Deep-dive novel samples; publish intelligence

1.6 Career Opportunities

Malware analysis sits at the heart of defensive security, and the demand is broad: security operations centers, incident-response consultancies, antivirus and EDR vendors, threat-intelligence teams, government and defense, and the internal security teams of any large organization. Because the skill is concrete and demonstrable, a portfolio of well-documented sample analyses (done safely, on public research samples) is one of the most persuasive things a beginner can show a hiring manager. Appendix E includes a career roadmap and practice resources to help you build one.

> **DID YOU KNOW?**
> Many professional analysts started exactly where you are — with a free hypervisor, a handful of open-source tools, and public research samples. Everything in this book can be practiced without any paid software.

1.7 Ethics and Legal Considerations

Handling malware is a position of trust. The knowledge that lets you analyze a threat could, misused, cause harm, so the field holds itself to clear rules. The essentials are simple: work only in an isolated lab you are authorized to use; only obtain samples through legitimate channels; never deploy, distribute, or "test" malware against systems you do not own or have explicit written permission to test; respect the privacy of any data you encounter inside a sample; and follow your organization's and jurisdiction's laws without exception.

© 2026 Codelivly. All Rights Reserved.