A CODELIVLY PUBLICATION

Security Researcher's Handbook

From Reconnaissance to Vulnerability Discovery, Exploitation, Reporting & Responsible Disclosure

Learn. Practice. Hack. Defend.

PUBLISHED BY

CODELIVLY

Copyright

Security Researcher's Handbook *From Reconnaissance to Vulnerability Discovery, Exploitation, Reporting & Responsible Disclosure*

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 reviews and certain other noncommercial uses permitted by copyright law.

Educational purpose. This book is intended solely for education, authorized security research, and professional development. Every technique, workflow, and example in this book is presented so that readers can practice in environments they own or are explicitly authorized to test.

No unauthorized testing. Nothing in this book authorizes you to test, probe, scan, or exploit any system you do not own or do not have explicit, documented permission to assess. Accessing computer systems without authorization is illegal in most jurisdictions and can carry severe civil and criminal penalties. The reader is solely responsible for ensuring that any activity they undertake is lawful and authorized.

Responsible research. The authors and publisher advocate coordinated, responsible disclosure and the minimization of harm in all security research. Readers are expected to act ethically and professionally at all times.

Limitation of liability. The information in this book is provided "as is," without warranty of any kind, express or implied. The authors and Codelivly shall not be liable for any loss, damage, or legal consequence arising from the use or misuse of the information contained herein.

Third-party trademarks. All product names, logos, tools, and brands mentioned in this book are the property of their respective owners. Their use is for identification and educational purposes only and does not imply endorsement.

No guarantee of results. This book does not promise employment, income, bug bounty rewards, certifications, or any specific career outcome. Skill development depends on the reader's own practice and diligence.

First Edition, 2026

Published by Codelivly

Dedication

To the curious, the ones who cannot look at a system without wondering how it really works, and what happens when it doesn't.

To the security researchers who spend long nights chasing a single anomaly until it gives up its secret. To the defenders who turn those discoveries into safer software for everyone. To the ethical hackers who choose the harder, honest path of asking permission and disclosing responsibly. And to every learner who refuses to accept "it just works" as an answer.

This book is for you.

Acknowledgements

A handbook like this is never the product of a single voice. It stands on decades of shared knowledge built by a global community that has chosen, again and again, to teach rather than hoard.

We thank the **readers** and **Codelivly learners** whose questions shaped every chapter. Your confusion told us where to slow down; your curiosity told us where to go deeper.

We are grateful to the broader **security research community**, the people who publish write-ups, file advisories, maintain vulnerability databases, and answer questions from strangers for no reward other than a stronger ecosystem. Modern security knowledge is overwhelmingly a gift economy, and this book is only possible because so many have given freely.

We thank the **open-source contributors** who build and maintain the tools that researchers rely on daily, and the **developers** who write the software we study, including those who respond to security reports with grace and fix issues that make the internet safer for everyone.

We thank the **bug bounty researchers** who have professionalized coordinated disclosure, the **educators** who make hard topics learnable, and the many **security communities**, conferences, and local meetups that welcome newcomers and keep the field honest.

Finally, we thank everyone who treats security research as a craft: patient, rigorous, and responsible. This book is our attempt to pass that craft forward.. The Codelivly Team

Foreword

Security research is, at its core, the disciplined study of how systems fail.

Every piece of software is a set of assumptions frozen into code. The developer assumed input would look a certain way. The architect assumed a component could be trusted. The operator assumed a default was safe. Most of the time those assumptions hold, and the system behaves. Security research is the practice of finding the places where they don't, and of doing so carefully, lawfully, and in a way that leaves the world more secure than it was before.

This matters more every year, not less. The systems we depend on are no longer single programs on single machines. They are sprawling compositions of web front-ends, mobile apps, APIs, third-party services, cloud infrastructure, identity providers, message queues, and open-source dependencies pulled from across the world. Each component is reasonable on its own. It is in the *seams* between them. The mismatched assumptions, the trust that was granted too generously, the edge case nobody owned, that real vulnerabilities live. The attack surface of a modern system is not a wall to be scaled; it is a vast, interconnected terrain to be mapped, understood, and questioned.

That is why this book insists, from its first pages, that research is a way of thinking before it is a set of tools. Tools change. Frameworks rise and fall. The vulnerability class that dominates headlines this year will be partly mitigated by better defaults next year, and replaced by something subtler. What endures is methodology: the ability to map an attack surface, form a precise hypothesis, investigate behavior patiently, validate a finding without causing harm, and reason honestly about impact. A researcher who has those habits can walk up to a technology they have never seen and make progress. A researcher who has only memorized commands cannot.

There is a second skill the field quietly demands, and it surprises many newcomers: communication. A vulnerability that cannot be reproduced does not exist, practically speaking. A finding that cannot be explained clearly to the engineer who must fix it will not get fixed. The researchers who make the greatest difference are not only those who find the deepest bugs, but those who can prove them cleanly, write them up precisely, and work with vendors constructively through the awkward, human process of

disclosure. This book treats reporting and disclosure as first-class research skills, because they are.

Finally, a word on responsibility. The same curiosity that makes a great researcher can, pointed carelessly, cause real damage to real people. The difference between a security researcher and an attacker is not skill. It is consent, restraint, and intent. Throughout this book you will be reminded, perhaps more often than you'd like, to work only in environments you own or are authorized to test, to minimize impact, and to disclose responsibly. That repetition is deliberate. It is the ethical spine of the profession.

If you read this book actively, building the labs, doing the exercises, keeping the research journal, you will come away not with a bag of tricks, but with a way of seeing. You will look at a login flow and wonder what happens at the boundaries. You will look at an API and ask who is really allowed to call it. You will look at a crash and ask whether it is merely a bug or a door. That shift in perception is the beginning of becoming a researcher.

Welcome to the work.

, The Editors, Codelivly

Preface

Why this book exists

Most people who want to become security researchers get stuck in the same place. They learn what a vulnerability class is. They can recite the definition of SQL injection or SSRF, and they can run the popular tools. But when they sit in front of a real, authorized target, nothing happens. The scanner finishes, reports a few low-severity issues, and they are left staring at an application with no idea what to actually *do*.

The gap is not knowledge of vulnerability names. The gap is **methodology**: the structured, repeatable way experienced researchers move from "here is a system I know nothing about" to "here is a validated, well-documented finding with a clear impact." That methodology is rarely written down, because for most practitioners it is tacit, absorbed over years through mentorship, CTFs, failed attempts, and reading hundreds of write-ups.

This book is an attempt to make that methodology explicit, teachable, and practiced.

Who should read this book

This book is written for readers who already have the basics and are ready to build professional research skills:

- You are comfortable on the Linux command line.
- You understand basic networking (IP, DNS, TCP, HTTP).
- You have seen the fundamentals of web security.
- You can read and write a little code, enough to follow Python and shell scripts, and to modify them.

You do **not** need prior professional experience in vulnerability research. We assume you are an intermediate learner, a student, a developer moving toward security, a penetration tester who wants to do original research, or a self-taught enthusiast who wants a structured path. If you are a complete beginner to computing, this book will be hard going; spend time with Linux, networking, and a programming language first, then return.

What you will learn

By the end of this book you will be able to:

- Map the attack surface of an authorized target from scattered public and in-scope information.
- Enumerate functionality and identify where trust boundaries lie.
- Form precise research hypotheses and investigate them methodically.
- Recognize, investigate, and safely validate the major vulnerability classes across web, API, authentication, mobile, cloud, source code, and binaries.
- Reason about more complex, logic-driven vulnerabilities that scanners miss entirely.
- Build minimal, reproducible, non-destructive proofs of concept.
- Determine real-world impact and severity honestly.
- Write professional vulnerability reports that get taken seriously and fixed.
- Disclose findings responsibly and coordinate with vendors.
- Turn your work into a portfolio and progress toward a security research career.

Above all, you will learn to **think like a researcher**.

How the book is organized

The book moves deliberately from foundations to advanced work.

- **Chapters 1 to 2** establish the mindset and the lab.
- **Chapters 3 to 4** teach reconnaissance and how to understand a web application deeply.
- **Chapters 5 to 8** cover vulnerability discovery, APIs, identity, and source-code analysis.
- **Chapters 9 to 11** move into reverse engineering, fuzzing, and binary exploitation concepts.
- **Chapters 12 to 14** address mobile, cloud, and the harder classes of vulnerability.
- **Chapters 15 to 17** cover proof-of-concept construction, professional reporting, and responsible disclosure.
- **Chapter 18** walks through realistic, fictional end-to-end case studies.
- **Chapters 19 to 20** focus on career development and present a complete, repeatable research playbook.

- The **appendices** provide a toolkit guide, printable checklists, reusable templates, a glossary, and curated directions for further learning.

How to use the labs and exercises

Every major chapter contains exercises, and they are the most important part of the book. They are deliberately *not* "type this command and watch the output." They ask you to investigate, form hypotheses, gather evidence, and reflect. Treat them as miniature research projects. You will learn ten times more from struggling through one exercise than from reading three chapters passively.

All practical work assumes a lab you control. Chapter 2 shows you how to build one. **Never** point the techniques in this book at systems you do not own or are not explicitly authorized to test.

Build a portfolio as you read

Throughout the book you will find **Portfolio Projects** and guidance on documenting your work. Do them as you go. By the time you finish, you can have a GitHub profile, a handful of write-ups, and a research journal that together demonstrate real capability to employers and the community, far more convincing than any certificate alone.

What this book does not attempt to teach

To stay focused and honest, this book deliberately leaves some things out. It is not a complete course in Linux, networking, or programming. You need those going in. It is not a tool manual; tools change too fast, and we teach methodology that outlives them. It is not a legal reference; laws vary by jurisdiction and we are not lawyers. And it is not a collection of ready-made exploits for real systems. That would be both unethical and counterproductive to your growth. Our goal is to make you a researcher, not to hand you someone else's answers.

How to Use This Book

Reading order

Read Chapters 1 and 2 first and in order, they set up the mindset and the lab that everything else depends on. After that, Chapters 3 through 17 build on one another and are best read in sequence, but experienced readers can jump to the domain they care about (API, mobile, cloud, reversing) once they have the foundations. Chapter 18's case studies make the most sense after you've met the vulnerability classes. Chapters 19 and 20, and the appendices, are reference material you will return to repeatedly.

Work in a lab, always

Set up the research environment from Chapter 2 before you attempt any practical exercise. The single most important rule in this book is simple: **practice only on systems you own or are explicitly authorized to test.** Intentionally vulnerable applications, local labs, and CTF platforms exist precisely so you can develop real skills without harming anyone or breaking the law. Use them.

Keep a research journal

From Chapter 1 onward you will maintain a **research journal**, a structured record of what you observed, what you hypothesized, what you tried, and what you found. The book introduces a consistent journal format and uses it throughout. This is not busywork. A disciplined journal is how researchers avoid repeating dead ends, how they build reproducible findings, and how a messy investigation becomes a clean report. It is also, over time, the raw material of your portfolio.

A minimal journal entry looks like this:

```
Research Question :
Target / Lab      :
Scope             :
Initial Observation:
Hypothesis        :
Evidence          :
Experiments       :
Results           :
Unexpected Behavior:
Potential Vulnerability:
Validation        :
Impact            :
PoC               :
Remediation       :
Final Conclusion  :
```

You will see this structure again and again. Adopt it early.

Take notes the way researchers do

Keep notes in plain text or Markdown, under version control (Chapter 2 shows a Git workflow). Capture raw evidence, requests, responses, logs, screenshots, alongside your interpretation, but keep the two visually separate so you never confuse what you *saw* with what you *concluded*.

Build the lab, record findings, write reports, build a portfolio

The rhythm of this book is a loop you will repeat for the rest of your career: build or extend your lab, investigate, record findings in your journal, turn confirmed findings into clean reports, and promote the best of that work into a public portfolio. Each chapter feeds that loop. If you follow it you will finish the book not just knowing *about* security research, but actually doing it.

Table of Contents

Dedication 4
Acknowledgements 5
Foreword 6
Preface 8
How to Use This Book 11

CHAPTER 1 What Is Security Research? 23
- Learning Objectives 23
- Introduction 23
- Core Concepts 24
- The Research Disciplines 26
- Ethics, Authorization, and the Law 30
- From Observation to Hypothesis 32
- The Research Lifecycle 33
- Case Study 35
- Practical Exercise 36
- Common Mistakes 38
- Researcher's Mindset Checklist 38
- Key Takeaways 39
- Questions for Review 40
- Portfolio Project 40

CHAPTER 2 Building the Security Research Environment 41
- Learning Objectives 41
- Introduction 41
- Core Concepts 42
- Recommended Lab Architecture 43
- Tool Categories 46
- Setting Up the Workbench 47
- Evidence Collection 50
- Handling Untrusted and Potentially Malicious Samples 51
- Research Workflow 52
- Case Study 52
- Practical Exercise 53

Common Mistakes ... 55
Researcher Checklist ... 55
Key Takeaways ... 56
Questions for Review ... 56
Portfolio Project ... 57

CHAPTER 3 Reconnaissance and Attack Surface Discovery ... 58
Learning Objectives ... 58
Introduction ... 58
Core Concepts ... 59
Reconnaissance Workflow ... 61
Technical Deep Dive: DNS and Certificates as Attack Surface ... 64
Turning Scattered Information into an Attack-Surface Map ... 66
Research Worksheet: Reconnaissance ... 67
Case Study ... 67
Practical Exercise ... 68
Common Mistakes ... 70
Researcher Checklist ... 70
Key Takeaways ... 71
Questions for Review ... 71
Portfolio Project ... 72

CHAPTER 4 Web Application Research ... 73
Learning Objectives ... 73
Introduction ... 73
Core Concepts ... 74
The Shapes of Web Interfaces ... 76
Practical Methodology ... 78
Research Workflow ... 79
Case Study ... 79
Practical Exercise ... 80
Common Mistakes ... 81
Researcher Checklist ... 81
Key Takeaways ... 81
Questions for Review ... 82
Portfolio Project ... 82

CHAPTER 5 Finding Vulnerabilities ... 83

Learning Objectives ... 83
Introduction ... 83
A General Methodology for Discovery ... 84
The Major Vulnerability Classes ... 85
Practical Methodology ... 91
Research Workflow ... 92
Case Study ... 92
Practical Exercise ... 93
Common Mistakes ... 93
Researcher Checklist ... 94
Key Takeaways ... 94
Questions for Review ... 94
Portfolio Project ... 95

CHAPTER 6 API Security Research ... 96
Learning Objectives ... 96
Introduction ... 96
Core Concepts ... 97
API-Specific Vulnerability Patterns ... 99
GraphQL in Depth ... 102
API Research Methodology ... 103
API Testing Checklist ... 104
Case Study ... 104
Practical Exercise ... 105
Common Mistakes ... 106
Researcher Checklist ... 107
Key Takeaways ... 107
Questions for Review ... 108
Portfolio Project ... 109

CHAPTER 7 Authentication, Identity, and Access Control Research ... 110
Learning Objectives ... 110
Introduction ... 110
Core Concepts ... 111
Federation: OAuth, SSO, and SAML ... 113
JSON Web Tokens (JWT) ... 115
Authorization Models and Privilege Boundaries ... 116
Account-Takeover Research Methodology ... 116