Preface

"I know the human being and fish can coexist peacefully."

- George W. Bush, Saginaw, Mich., Sept. 29, 2000

Introduction

The concept of fuzzing has been around for almost two decades but has only recently captured widespread attention. A plague of vulnerabilities affecting popular client-side applications, including Microsoft Internet Explorer, Microsoft Word, and Microsoft Excel, were seen in the year 2006 and a large portion of these vulnerabilities were discovered through fuzzing. This effective application of fuzzing technologies gave rise to new tools and increased exposure. The sheer fact that this is the first published book dedicated to the subject matter is an additional indicator that there is an increasing interest in fuzzing.

Having been involved in the vulnerability research community for years, we have used a variety of fuzzing technologies in our day to day work, ranging from immature hobby projects to high end commercial products. Each of the authors has been involved in the development of both privately held and publicly released fuzzers. We leveraged our combined experience and ongoing research projects to compose this bleeding edge book, which we hope you will find beneficial.

INTENDED AUDIENCE

Security books and articles are frequently written by security researchers for the benefit of other security researchers. We strongly believe that the quantity and severity of vulnerabilities will continue to grow so long as security is deemed to be the sole responsibility of a security team. As such, we have taken strong efforts to write for a large audience including both readers who are new to fuzzing and those who have already had significant exposure.

It is unrealistic to believe that secure applications can emerge from the development process if we simply hand completed applications to a security team for a quick audit prior to production launch. Gone are the days when a developer or a member of the QA Team can say, "security's not my problem – we have a security team that worries about that". Security must now be everyone's problem. Security must be baked into the software development lifecycle (SDLC), not brushed on at the end.

Asking the development and QA teams to focus on security can be a tall order, especially for those that have not been asked to do so in the past. We believe that fuzzing presents a unique vulnerability discovery methodology that is accessible to a wide audience due to the fact that it can be highly automated. While we are hopeful that seasoned security researchers will gain valuable insights from this book, we are equally hopeful that it will be accessible to developers and QA teams. Fuzzing can and should be an integral part of any SDLC, not just at the testing phase, but also during development. The earlier a defect can be identified, the less costly it will be to remediate.

PREREQUISITES

Fuzzing is a vast subject. While we cover many non-fuzzing specific basics throughout the book, a number of assumptions regarding prior knowledge have been made. Readers should have at least a basic understanding of programming and computer networking prior to taking on this book. Fuzzing is all about automating security testing so naturally much of the book is dedicated to building tools. We have purposely selected multiple programming languages for these tasks. Languages were selected according to the task at hand, but this also demonstrates that fuzzing can be approached in numerous ways. It is certainly not necessary to have a background in all of the languages used, but having a language or two under your belt will go a long way in helping you to get the most from these chapters.

We detail numerous vulnerabilities throughout the book and discuss how they might have been identified through fuzzing. However, it is not our goal to define or dissect the nature of the vulnerabilities themselves. Many excellent books have been written which are dedicated to this topic. If you are looking for a primer on software vulnerabilities,

Exploiting Software by Greg Hoglund and Gary McGraw, books from the Hacking Exposed series and *The Shellcoder's Handbook* by Jack Koziol, David Litchfield, et al. are great references.

APPROACH

How to best leverage this book depends upon your background and intentions. If you are new to fuzzing, we would recommend digesting the book in a sequential manner as it has been intentionally laid out to provide necessary background information prior to moving onto more advanced topics. If however, you've already spent time using various fuzzing tools, don't be afraid to dive directly into topics of interest as the various logical sections and chapter groupings are largely independent of one another.

Part I is designed to set the stage for the specific types of fuzzing that are discussed in the remainder of the book. If you're new to the world of fuzzing, consider this to be required reading. Fuzzing can be used as a vulnerability discovery methodology for just about any target, but all approaches follow the same basic principles. In Part I, we seek to define fuzzing as a vulnerability discovery methodology and detail the knowledge that will be required regardless of the type of fuzzing which is conducted.

Part II focuses on fuzzing specific classes of targets. Each target is divided across two or three chapters. The first chapter provides background information specific to the target class and the subsequent chapters focus on automation, detailing the construction of fuzzers for that particular target. Two automation chapters are provided when separate tools are deemed necessary for the Windows and UNIX platforms. For example, consider the chapter triplet on "File Format Fuzzing" starting with Chapter 11 which details background information related to fuzzing file parsers. Chapter 12, "File Format Fuzzing: Automation on UNIX" details the actual programming of a UNIX-based file fuzzer and Chapter 13, "File Format Fuzzing: Automation on Windows" details the construction of a file format fuzzer designed to run in the Windows environment.

Part III tackles advanced topics in fuzzing. For readers who already have a strong knowledge of fuzzing it may be appropriate to jump directly into Part III, while most readers will likely want to spend time in Parts I and II before moving onto these topics. In Part III, we focus on emerging technologies that are just beginning to be implemented but will become critical for advanced vulnerability discovery tools that leverage fuzzing in the future.

Finally, in Part IV, we reflect on what we've learned throughout the book and then peer into the crystal ball to see where we're headed in the future. While fuzzing is not a new concept, it still has plenty of room to grow, and we hope that this book will inspire further research in this space.

A Touch of Humor

Writing a book is serious work, especially a book on a complex subject like fuzzing. That said, we like to have fun as much as the next person (actually probably significantly more than the average person) and have made our best effort to keep the writing entertaining. In that spirit, we decided to open each chapter with a brief quotation from the 43rd President of the United States, George W. Bush (a.k.a. Dubya). No matter what your political affiliation or beliefs may be, no one can argue that Mr. Bush has cooked up many entertaining quotes over the years, enough to fill entire calendars[1] even! We've compiled some of our favorites to share with you and hope you find them as funny as we do. As you'll see throughout the book, fuzzing can be applied against a variety of targets, evidently even the English language.

About the Cover

At times, vulnerabilities have at times been referred to as "fish." (For example, see the thread on "The L Word & Fish"[2] from the DailyDave security mailing list.) This is a useful analogy that can be applied across the board when discussing security and vulnerabilities. A researcher can be called a fisherman. Reverse engineering the assembly code of an application, line by line, in search of a vulnerability may be referred to as "deep sea fishing." In comparison to many other auditing tactics, fuzzing for the most part only scratches the surface and is highly effective at capturing the "easy" fish. In addition, the grizzly bear is a notable "fuzzy," yet powerful animal. Combined, these are the main motivations behind our choice of cover art where the bear, representing a fuzzer, is shown capturing a fish, representing a vulnerability.

Companion Website: www.fuzzing.org

The fuzzing.org website is an absolutely integral part of this book as opposed to a supplemental resource. In addition to housing errata that is sure to emerge post publication, the website will serve as the central repository for all source code and tools covered throughout the book. Over time, we intend to evolve fuzzing.org beyond a book-centric companion website into a valuable community resource with tools and information related to all fuzzing disciplines. We welcome your feedback in order to help make the site a valuable and open knowledgebase.

---

1 http://tinyurl.com/33l54g

2 http://archives.neohapsis.com/archives/dailydave/2004-q1/0023.html