1. Record Nr. UNINA9910458833603321

Titolo	Open source fuzzing tools [[electronic resource] /] / Gadi Evron ... [et al.]
Pubbl/distr/stampa	Burlington, MA, : Syngress Pub., c2007
ISBN	1-281-14515-7 9786611145156 0-08-055561-6
Edizione	[1st edition]
Descrizione fisica	1 online resource (209 p.)
Altri autori (Persone)	EvronGadi
Disciplina	005.14 005.8
Soggetti	Computer software - Testing Open source software Debugging in computer science Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Includes index.
Nota di contenuto	Front Cover; Open Source Fuzzing Tools; Copyright Page; Contributing Authors; Contents; Chapter 1: Introduction to Vulnerability Research; Statement of Scope; Off-by-One Errors; Programming Language Use Errors; Integer Overflows; Bugs and Vulnerabilities; The Vaunted Buffer Overflow; Finding Bugs and Vulnerabilities; Source Code Review; Black Box Testing; Glass Box Testing; Chapter 2: Fuzzing-What's That?; Introduction; Introduction to Fuzzing; Milestones in Fuzzing; Fuzzing Technology; Traffic Sniffing; Prepared Template; Second-Generation Fuzzing; File Fuzzing; Host-side Monitoring Vulnerability Scanners as Fuzzers Uses of Fuzzing; Open Source Fuzzers; Commercial-Grade Fuzzers; What Comes Next; The Software Development Life Cycle; Chapter 3: Building a Fuzzing Environment; Introduction; Knowing What to Ask...; Basic Tools and Setup; Data Points; Crash Dumps; Fuzzer Output; Debuggers; Recon Tools; Linux; OSX; Summary; Chapter 4: Open Source Fuzzing Tools; Introduction; Frameworks; Special-Purpose Tools; General-Purpose Tools; Chapter 5: Commercial Fuzzing Solutions; Introduction; beSTORM (by Beyond

Security); BPS-1000 (by BreakingPoint Systems); Codenomicon Mu-4000 Security Analyzer (by Mu Security)Chapter 6: Build Your Own Fuzzer; Hold Your Horses; Fuzzer Building Blocks; One or More Valid Data Sets; Understanding What Each Bytein the Data Set Means; Change the Values of the Data Sets While Maintaining the Integrity of the Data Being Sent; Recreate the Same Malformed DataSet Time and Time Again; An Arsenal of Malformed Values, or the Ability to Create a Variety of Malformed Outputs; Maintain a Form of a State Machine; Summarize; Down to Business; Simplest Fuzz Testing Find Issues; Chapter 7: Integration of Fuzzing in the Development Cycle Introduction Why Is Fuzzing Important to Include in a Software Development Cycle?; Security Testing Workload; Setting Expectations for Fuzzers in a Software Development Lifecycle; Fuzzing as a Panacea; Fuzzing Tools versus ...; Setting the Plan for Implementing Fuzzers into a Software Development Lifecycle; Setting Goals; Building and Executing on the Plan; Understanding How to Increase Effectiveness of Fuzzers, and Avoiding Any Big Gotchas; Hidden Costs; Finding More Vulnerabilities; Summary; Solutions Fast Track; Frequently Asked Questions; Chapter 8: Standardization and Certification Fuzzing and the Corporate Environment Software Security Testing, the Challenges; Testing for Security; Fuzzing as a Viable Option; Business Pressure; Software Security Certification; Meeting Standards and Compliance; Tester Certification; Industry Pressure; Antivirus Product Testing and Certification; Chapter 9: What Is a File?; Introduction; Are File Fuzzers Special?; Analyzing and Building Files; Textual Files; Binary Files; Running the Test; Monitoring the Application with the Test Cases; Chapter 10: Code Coverage and Fuzzing; Introduction; Code Coverage; Obtaining Code Coverage Instrumenting the Binary

Sommario/riassunto

A ""fuzzer"" is a program that attempts to discover security vulnerabilities by sending random data to an application. If that application crashes, then it has defects to correct. Security professionals and web developers can use fuzzing for software testing--checking their own programs for problems--before hackers do it! Open Source Fuzzing Tools is the first book to market that covers the subject of black box testing using fuzzing techniques. Fuzzing has been around for a while, but is making a transition from hacker home-grown tool to commercial-grade quality assurance