Industrial Cybersecurity Training programs

Vulnerability discovery through fuzzing

Fuzzing is a powerful and effective technique for finding vulnerabilities in software. Despite the best efforts of security researchers and software developers, complex software still ships with bugs. Fuzzing helps to discover hard-to-find vulnerabilities that may not be caught by manual code review, static code analyzers, or unit tests.

Each year, thousands of vulnerabilities are responsibly disclosed by security researchers. By using fuzzing, you can join their ranks and help make the software we use every day more secure and reliable.

With fuzzing, you can also tailor your testing to your specific needs and requirements. You can use different fuzzing tools and techniques to target specific parts of your software, or to test for specific types of vulnerabilities. This flexibility and customization make fuzzing a valuable addition to any software testing toolkit.

Learning objectives

Vulnerability discovery through fuzzing aims to introduce professionals involved in defining, developing, and testing software products to the practice of fuzzing. This training will show attendees how to use fuzzing to identify vulnerabilities in their own software, as well as in third-party products, with or without access to the source code.

During the training, attendees will learn about the fundamental concepts of fuzzing, including corpus mutation, coverage, feedback, and instrumentation. We will also discuss the importance of these concepts and how they apply to the practice of fuzzing.

In addition to learning the theoretical aspects of fuzzing, attendees will also have the opportunity to apply their knowledge in hands-on exercises using real-world examples. These examples will illustrate the concepts discussed in the training and will cover both white box and black box scenarios.

By the end of the training, attendees will have the skills and knowledge they need to use fuzzing effectively to discover vulnerabilities.

Course topics

- Introduction to fuzzing techniques, how and when to use it
- Writing a custom fuzzer
- Getting hands-on existing tools such as libfuzzer, AFL++, DynamoRIO, WinAFL
- Conducting fuzzing against both source code and binary targets
- Emulation for fuzzing other architectures
- Effective corpus generation and mutation
- Crash analysis and triaging

Training prerequisites

- Being able to read and write some snippets of C/C++ code
- Familiarity with basic Linux commands
- Basic reverse engineering skills
- Experience using a disassembly tool preferable but not mandatory

Course format

Onsite instructor led lessons, workshop, hands-on sessions. A hybrid or online format is possible

Duration

3 days

Group

Up to 10 people

Who can benefit:

- Software engineers
- QA testers
- Security researchers
- Developers
- Penetration testers

Resources

- Course materials
- Lab manuals