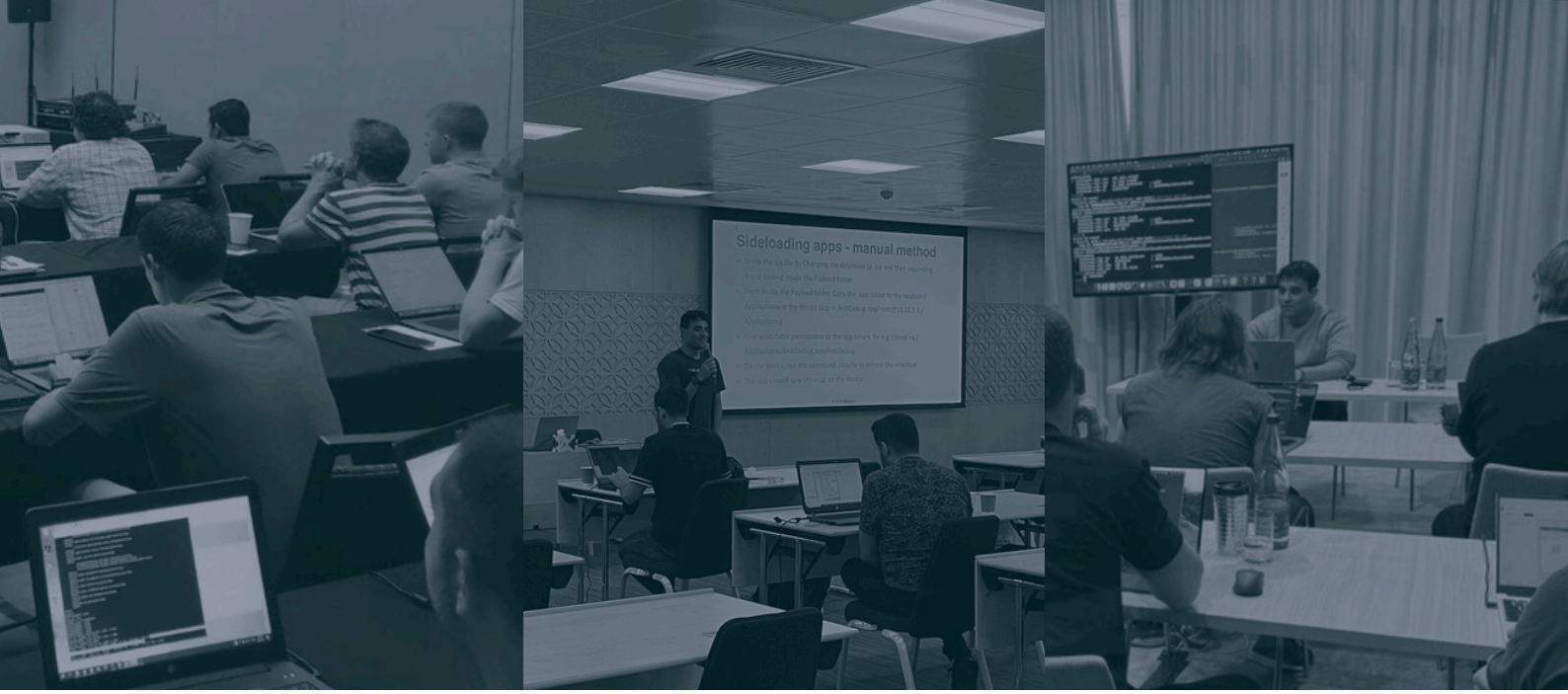


APPLIED FUZZING AND VULNERABILITY ANALYSIS

Expert-Led Cybersecurity Training · Beginner to Advanced

COURSE OVERVIEW

This training empowers you to harness the power of fuzzing, an automated technique that uncovers hidden vulnerabilities in software. Manual testing for these weaknesses in complex codebases is a struggle. Fuzzing automates this process, feeding your software unexpected inputs to expose cracks in its armor. By integrating fuzzing into your Secure Development Lifecycle (SDLC), you can proactively identify and fix vulnerabilities early, saving time and resources down the line. This training equips you with the knowledge to not only understand fuzzing fundamentals but also apply them across various platforms like Linux and Windows. You'll gain expertise in triage analysis, allowing you to prioritize and effectively address the vulnerabilities identified through fuzzing. Through hands-on labs, you'll gain real-world experience with the "Crash, Detect & Triage" process, solidifying your fuzzing mastery. This training is designed for security professionals and developers who want to take a proactive approach to software security.

8KSEC

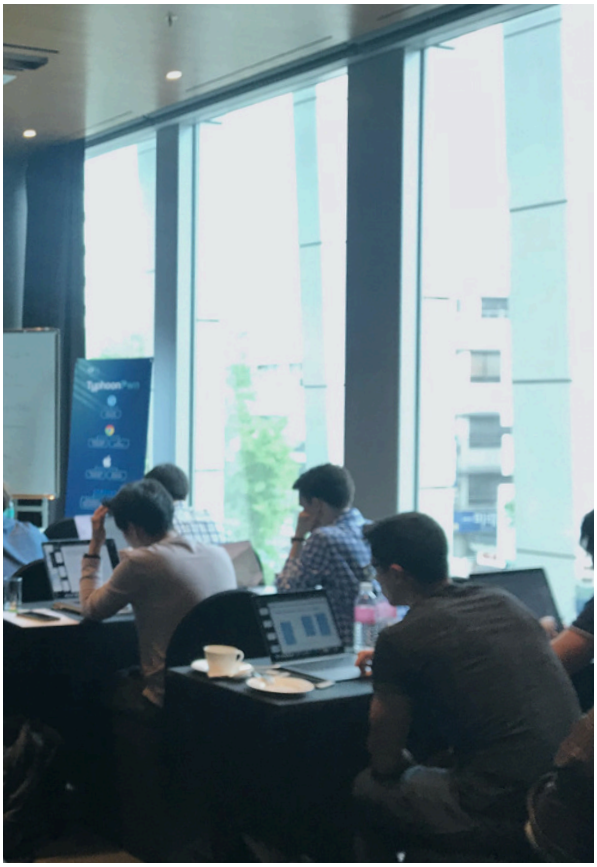
KEY LEARNING OBJECTIVES

- Efficient fuzzing techniques
- Exploring various vulnerability classes
- Essential basics and mechanics of fuzzing
- Designing custom grammars for fuzzing
- Establishing persistence in intricate programs
- Leveraging QEMU for binary-centric fuzzing
- ARM architecture introduction and ARM binary fuzzing
- Initiating fuzzing for Windows binaries
- Numerous practical exercises with real-world software
- CTC - Capturing crashes in custom applications

WHY SHOULD YOU TAKE THIS COURSE?

This is a completely hands-on course designed for beginners and intermediate students. Students will discover advanced fuzzing techniques, understand various vulnerabilities, create custom grammars, test complex programs, use QEMU & ARM for fuzzing, tackle Windows binaries, practice with real-world exercises, and master crash capture challenges. This training is designed in such a way that it introduces the concept of fuzzing and vulnerability discovery in software's covering multiple platforms such as Linux & Windows and triage analysis for those vulnerabilities.

(Continued on the next page)



PREREQUISITE KNOWLEDGE

- Working knowledge of cybersecurity and pentesting fundamentals
- Working knowledge of Fuzzing concepts and Corpus generation is recommended, but not required
- Basic Windows & Linux skills and command-line proficiency
- Understanding of fundamental programming concepts and looping structures in at-least one higher-level language
- Basic Windows/Linux binary assembly knowledge is recommended, but not required

WHAT WILL THE STUDENTS GET?

- Training Manual for the course
- A dedicated server with custom OS (Windows & Linux) for one month
- Lab setup (OVA of Ubuntu and Windows) loaded with all the course
- Exercise material including solutions to all of the exercises
- A private dedicated channel where trainers will be available to answer your queries after the training

WHO SHOULD ATTEND?

This training program is designed for individuals and professionals seeking to acquire a comprehensive understanding of the fundamentals of fuzzing.

HARDWARE/SOFTWARE REQUIREMENT

- Laptop with a minimum of 6GB RAM and 40GB free hard disk space
- VMware Workstation, VMware Fusion (even trial versions can be used) or VirtualBox.
- You must have full administrator access to the Windows operating system installed inside the VMware Workstation/Fusion.

Detailed Course Setup instructions and Slack access will be sent a few weeks prior to the class

(Continued on the next page)

COURSE SYLLABUS

Module 1: Fundamentals of Fuzzing

- Understanding fuzzing fundamentals
- AFL++ Internals
- Setting up the environment
- Selecting fuzzing targets
- Spinning up the fuzzer effectively
- Corpus generation
- Address/Memory Sanitizers
- Hooking custom mutators
- Parallel fuzzing
- Improving code coverage with grammar
- Plotting difference in code coverage
- Enhancing your fuzzing approach

Module 2: Fuzzing and Crash Analysis

- Setting up persistent mode
- Introduction to QEMU
- AFL++ QEMU mode
- Targeting blackbox binaries
- Introduction to ARM
- Cross-platform architecture fuzzing
- Setting up QEMU persistent
- Introduction to network fuzzing
- WinAFL++ Internals & modern alternatives (Jackalope, AFL++/WinAFL2)
- Fuzzing windows binaries
- Analyzing your target with debuggers
- Improving code coverage

Module 3: Advanced Fuzzing Techniques

- Symbolic execution fuzzing
- Introduction to libFuzzer
- Writing simple libFuzzer harness
- Setting up ClusterFuzz (& other fuzzers: OSS-Fuzz, Honggfuzz, Nyx-Fuzz, Snapchange)
- Fuzzing browser engines and SSL libraries
- Overview of different fuzzing frameworks
- Integrating slack with fuzzing stats
- Capture the crash
- Structure-aware / grammar-based fuzzing (e.g., libprotobuf-mutator, Nautilus)
- Snapshot-based fuzzing (e.g., Nyx-Fuzz, Snapchange)

About the company

8kSec is a foremost cyber security research company offering exceptional training and consulting services to aid clients in enhancing their security stance. Our experts possess extensive experience in delivering specialised cybersecurity training and consulting to several commercial and defence organisations across the United States, Europe, and the Middle East and North Africa region.

Get in touch

8kSec.io
trainings@8ksec.io