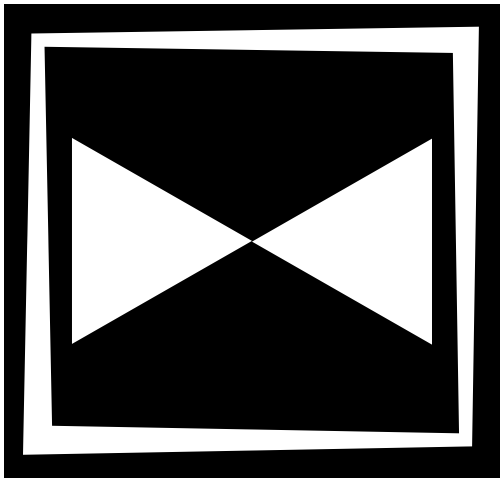

Zero Day Engineering

training & intelligence

ZERO DAY ENGINEERING

TRAINING OVERVIEW

This 4-day training course comprehends essential knowledge and skills required to conduct modern low-level security research, vulnerability exploit development, secure software engineering and hardening. The course is created and taught by a professional vulnerability researcher and low-level hacker who is actively engaged in security bounty hunting and zero day vulnerability discovery and exploitation as her main job, and qualified as a top zero day hacker by successful participation in Pwn2Own competitions.

This course is designed for complete beginners, yet inclusive of many advanced aspects of the craft that students will likely need as they advance. Following the always systematical structure of Zero Day Engineering Project courses, it covers all the essentials that students need to get started with arbitrary research targets in a compact and practical way. Excluded are specialized topics: advanced exploit techniques and mitigations, OS & hardware theory, and threat models for specific software classes.

Course materials will be illustrated and practiced with modern and immediately relevant real-life case studies, while omitting toy examples that entry level courses commonly rely on.

AUDIENCE

This training course is primarily intended for beginners in low-level security research with basic knowledge in computer science.

Materials of this course comprehend an essential base of knowledge and skills for software vulnerability researchers, security-minded software developers, and DevSecOps engineers.

Systematical structure and strategic insights of this course would be valuable for seasoned hackers and professional security researchers.

PREREQUISITES

Mandatory:

- General computer science.
- A laptop with a modern Linux VM.

Optional.

- C-language.
- Familiarity with basic concepts of low-level computer security.

OBJECTIVES

Upon completion of this training course the students should – with a certain commitment – be able to:

- Engage in bug bounty hunting.
- Participate in CTF competitions.
- Conduct exploit & vulnerability analysis.
- Find simple 0day vulnerabilities.
- Reverse-engineer simple softwares..
- Avoid security bugs in their code.
- Write and customize fuzzing tools.
- Apply public tools and scripts in the right order.
- Make independent and reasonably correct judgements about 3rd party bugs & exploits.
- Choose further specialization and research focus wisely.

LEVELS & CERTIFICATION

Complexity: beginners, involved.
Certificate grade: C0DE0 (Foundation).
Next level training: "<Target> Vulnerability Research"

TRAINING DETAILS

This training course assumes an extremely ambitious goal: to comprehend everything you need to get started with modern application security research in just 4 days. This is accomplished by following a deeply systematical learning model created by the instructor with an extensive experience in multiple specializations of the subject, alongside with time-proven techniques of cognitive optimization.

Despite the sassy name, the knowledge presented in this course is ethically neutral. It provides a universal foundation theory and skills for writing secure code, as much as for offensive security research of arbitrary software, firmware and hardware targets. Students are encouraged to think generically in terms of universal models, that makes a solid ground for arbitrary specializations: such as bug hunting, writing fuzzing tools, exploit development, reverse-engineering, or malware analysis.

The course deliberately ignores any differences between attacking binary code and web applications, because the underlying threat models are essentially the same, as explained on Day 1. That said, course approach is heavily geared towards binary analysis. Even logic vulnerabilities – the bug class which is commonly associated with web applications – are considered with respect to modern binary programs.

Key principle is to expose attendees immediately to some of the most impactful real-life case studies. While it may be impossible to learn it all in a few days, eager students are expected to be able to enter CTF competitions and find some easy bugs straight out of the training. In addition, the solid foundation given by this course will later help them to learn specialized knowledge faster and more systematically.

An empirical sampling of various specializations in the field of application security, alongside with insights of a seasoned hacker, comes as a side effect in this course. That would enable newcomers to choose their specialization based on personal experience rather than theoretical reasoning or fashion.

Course experience is a mixture of theory and practice. Practicals – that's small exercises and labs – are based on Linux, as the most versatile technical research platform that every low-level hacker eventually employs extensively in their workflow, even if their research is focused on a different OS.

The knowledge presented in this course is 100% original, based on independent technical research and reverse engineering work done by the author herself, and illustrated with security vulnerabilities discovered by the author, as well as with publicly available sources. Course content is kept up-to-date.

ABOUT THE INSTRUCTOR

Alisa Esage is a lifetime code breaker, vulnerability researcher and reverse engineer. She was credited by Microsoft, Google, Firefox, Oracle, Schneider Electric, and other leading software vendors for discovery of previously unknown security bugs. She is specialized on low-level offensive security research and exploit development for hardened systems.

Alisa taught herself assembly programming and reverse-engineering while studying in school. She is active in 0day vulnerability research since 2011, is deeply familiar with key research directions in the field of low-level offensive security and exploitation, while leaning to generalization and abstract modeling. Alisa is qualified as a top zero day hacker by successful participation in Pwn2Own competitions.

PROGRAM AT A GLANCE

DAY 1. FOUNDATION

Abstract models and essential general theory.

- Program theory concepts.

System design, program state space, code correctness.

- Model of Vulnerability.

Incorrect code behavior and weird program states. Bug vs. vulnerability. Root cause vs. effect.

- Model of Exploit.

Objectives, stages, primitives, building blocks.

- General vulnerability classification.

Design bugs, logic bugs, memory safety bugs.

- Application Threat Models.

Attack surfaces & attack vectors in software applications.

- C and C++.
- JavaScript.
- Python.

DAY 2. PROGRAM ANALYSIS & LOGIC BUGS

Working with source codes. Finding, evaluating and reproducing simple types of bugs.

- Static code analysis.

Bug hunting in large and complex code bases. Manual skills vs. automation.

- Assisted code analysis.

Finding bugs in code with modern tools. CodeQL.

- Logic bugs.

Classification and common bug patterns.

- Case studies.

High-impact logic bugs in modern software. Binary targets vs. web apps.

- Security patch analysis.

Understanding bugs via source code patches.

- Introduction to dynamic analysis.

Debuggers in source mode. Frida.

DAY 3. WORKING WITH BINARY CODE

Everything you need to get started.

- Compiler theory.

From HLL to binary code.

- Assembly programming.

CPU models & popular architectures.

- Reverse code engineering.

From binary to assembly to HLL to program architecture and design.

- Memory safety bugs.

Classification & common patterns. Case studies.

- Introduction to fuzzing.

From random inputs to effective automated vulnerability discovery.

- Modern shellcoding.

From weird binary program state to CPU control.

DAY 4. INTO THE FUTURE

Up-and-coming essentials in low-level security.

- Memory-safe programming languages.

Rust. Internals, bugs, academic research & theory.

- Blockchain security.

New threat models & case studies.

- Hardware bugs.

Speculative execution, Rowhammer, FI.

- DSP and NPU.

Software, firmware and hardware that powers artificial neural networks and deep learning.

- Mathematics of low-level security.

Program modeling efforts, weird machines, abstract interpretation.

Note: most bullet point sections will include at least one hands-on practice.

TRAINING PACKAGES & FEES

This training course is publicly available online (live), and privately.

Online training is based on a modern streaming platform with high-quality audio and video, and a group chat. The instructor will be available to all students for questions, feedback and technical support.

The Basic and Advanced packages of the live online training differ by the instructor's level of involvement with your study.

All training packages were specifically optimized for online experience.

LIVE ONLINE TRAINING

Basic package

What's included:
- Access to public online training.
- Guaranteed instructor's feedback by email.
- Training slides and materials.
- Training completion certificate.

Price: €3,900.- per person.

Advanced package

What's included:
- Everything in the Basic package.
- Personal feedback and technical support from the instructor during the live training.
- A possibility to receive a Training Achievement certificate by undergoing an assessment.

Price: €4,200.- per person.
Limited number of seats.

SELF-PACED (NOT AVAILABLE)

Basic package

What's included:
- Video lectures, exercises, and walk-through.
- Training slides and materials.
- Training completion certificate.

Price: -

Complete package

What's included:
- Everything in the Basic package.
- One month of technical support by email.
- An on demand personal consultation with the instructor by video call.
- Join our online public training on the same topic during the year at no additional cost.

Price: -

PRIVATE & CUSTOMIZED TRAINING

Minimal private group size is 10 persons. Availability is limited, book at least 3 months in advance.

LIVE TRAINING DATES & BOOKING

Refer to our website for the dates of the nearest public training. This training would be primarily held on, and optimized for, ONLINE content delivery.

All our online training courses are offered exclusively at the Zero Day Engineering Project website: http://zerodayengineering.com.

Bookings of public training seats are accepted via the website. For custom and private bookings, email us.

WHAT TO EXPECT?

Quoted below are anonymized extracts of private feedback from the students of Zero Day Engineering online training courses.

"It was empowering. Not only did I feel like I learned an enormous amount, but by the end I felt confident I knew **how to start looking for real vulnerabilities in virtualization systems**."

"I had an amazing time in the training. I feel like a lot of the **knowledge I had was clarified in the training and is now more organized**. Of course I also **learned a lot of new stuff** and it was really interesting and useful."

"It is a well written training, both the materials/slides and exercises are all **well designed**. I also really appreciated the knowledge you showed in the training, it is clear you have **a lot of experience in hypervisor research** and it was great to learn from you."

"I feel like the fact that a big part of the training was to show **how to research and explain your methodology** was really good, it was useful to learn how to approach a problems/research objective when it comes to different attack vectors."

"I really like the **more technical parts** – e.g. different IO options, how hardware virtualization works, OS ABC, MMU virtualization, I found them more interesting than the **specifics of a certain hypervisor**. Also liked the part where you compare different vulnerability types, and how the type recent vulnerabilities indicate the kind of scrutiny a project has seen."

"[I learned that] finding bugs in virtualization systems is achievable. Before doing this course **hypervisor exploitation seemed like an unknowable thing** that was just "too hard". I don't have anyone in my professional network or friend groups that knows anything about it, and information online is scarce. Learning from your course, and especially performing the exercises, has given me the **confidence to dive in** and start looking for bugs."

"The **processes and workflows** that you demonstrated. Particularly during your walkthroughs of the exercises, it was incredibly valuable to see and hear **your own methodology** for completing each example. The exercises themselves where also a fantastic learning tool."

"Here is what I loved about the whole thing:
- Well organized content, with a good order of things.
- A decent **balance of theory and hands-on** (I'm probably biased to hands-on).
- Pomodoro, time boxing, neural net.. liked the **meta-learning** touch there.
- Discussions on threat models, vuln discovery strategies, potential fuzzing designs."

"Loved the 25 minutes exercises, **really intense and gets you involved**."

Some public reviews from our students can be found on our website.

FURTHER INQUIRIES

E-mail: contact@zerodayengineering.com
Note: we typically respond to all business e-mails within 1-2 business days. If you didn't receive a response, that may be due to a failure of e-mail systems or spam filters. Reserve contact is Twitter.

RESERVED FOR NOTES

This document was last updated on February 22nd, 2022 (initial release)