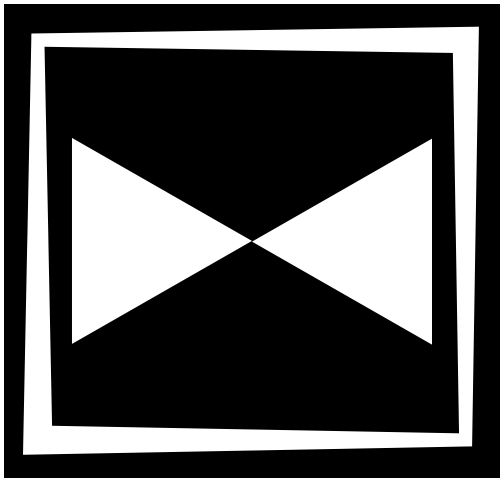

Zero Day Engineering

training & intelligence

Zero Day Engineering LLC
http://zerodayengineering.com

ADVANCED HYPERVISOR EXPLOITATION

TRAINING OVERVIEW

This 4-day training course is a full-stack exploit development bootcamp targeting hypervisors. It is created and taught by a highly experienced binary hacker who is specialized in hypervisor security research, having established credibility as a successful participant in Pwn2Own competitions in the Virtualization category, reverse engineered and found 0day bugs in multiple hypervisor implementations.

Key principle in this course is a systematical, broad and practical coverage of everything that one could possibly need to develop a high-quality exploit for an arbitrary hypervisor, given an exploitable bug. We'll take a look at various attack scenarios: from DoS to Infoleak to RCE and VM Escape, and various classes of security vulnerabilities that are common in hypervisors, with their respective exploitation techniques for both open source and proprietary software. Set of practical exercises will be targeting VMware ESXi.

For best results in the strategic perspective attendees are advised to take this training together with the "Hypervisor Vulnerability Research" training. Beginners will be assumed to take the latter training first.

AUDIENCE

Primary audience: professional application security researchers and low-level hackers.

System programmers and virtualization engineers will benefit from this course by gaining an advanced attacker's perspective on their code.

Those specialists who are already familiar with hypervisor security will find value in the original methodology and the all-encompassing knowledge exposition of this course.

PREREQUISITES

Mandatory:

- C-code reading & Python basic coding.
- x86_64 assembly and reverse engineering.
- VMware Fusion with nested virtualization.

Recommended (optional):

- "Hypervisor Vulnerability Research" training.
- Prior experience in exploit development.

OBJECTIVES

Upon completion of this training course students should be able to:

- Write an exploit for any hypervisor, either open source or proprietary.
- Quickly evaluate whether the given bug would be easy to exploit, and the time required.
- Choose the right exploitation technique and exploit engineering primitives for the given bug.
- Avoid reinventing well known exploitation techniques by being familiar with public work.
- Construct original exploits and primitives.
- Understand the relative technical complexities of popular hypervisor implementations with respect to exploit engineering.

LEVELS & CERTIFICATION

Complexity: advanced material.
Certificate grade: C0DE7 (ZDE Exploit Engineering).
Next level training: specialized on your target.

TRAINING DETAILS

This 4-day training course is designed to convey a comprehensive system of knowledge about advanced exploit engineering for arbitrary hypervisors and low-level virtualization systems. Attendees will be exposed to the peculiarities of developing exploits for hypervisors, including a broad variety of exploitation techniques for popular hypervisor implementations and bug classes, and the respective practical skills.

Training content is focused on exploit development, given an exploitable bug. That largely excludes vulnerability discovery and threat modeling considerations. The latter vulnerability research topics are covered extensively in our specialized "Hypervisor Vulnerability Research" course.

Hands-on work is 40-50% of the training, comprised of exercises and labs that represent various stages of engineering an advanced hypervisor exploit. Lectures include both essential theory and a review of specific vulnerability exploitation techniques.

Key principle is to expose systematically a broad spectrum of knowledge points that one is likely to need at some point when attacking hypervisors. We'll take a look at both logic bugs and memory safety issues, various attack scenarios from DoS to Infoleak to RCE and VM Escape, and various classes of vulnerabilities with their respective exploitation techniques. We'll then practice to write an advanced hypervisor exploit targeting open source code (initially) and then proprietary binary code.

Gradual rise of complexity is another principle that is encoded in the course structure. We start from relatively easy bug classes and exploit techniques that don't need advanced shell-coding skills, and wrap up with a theoretical exposition of modern state-of-the-art exploitation trends and emerging topics.

VMware ESXi was chosen to be our practical framework for purposes of exploit development exercises, as the most obscure and technically challenging enterprise-grade hypervisor today. Note, because this course strives to be universal and implementation-neutral, target system internals and reverse engineering will be discussed only to the extent as it is required by the practical case study. Eager advanced researchers are referred to our specialized "Reverse Engineering VMware ESXi" training.

All training materials are original and created by the author specifically for this course. The exploits on which exercises are based were developed by the author herself for prior publications or competitions.

ABOUT THE INSTRUCTOR

Alisa Esage is a professional low-level 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 attacking popular and hardened software and firmware implementations for exploit development.

For several years Alisa focused on modern virtualization security and system internals, working on discovery and exploitation of vulnerabilities in multiple popular hypervisors. In the process she developed a deeply generalized perspective on attacking modern hypervisors, which forms the basis of this training. Alisa is the winner of Pwn2Own 2021 competition in the Virtualisation category.

Version: 0.1

PROGRAM AT A GLANCE

DAY 1. WARMING UP WITH LOGIC BUGS

Lectures:

- Recap of theory.

Revise the relevant theoretical concepts and knowledge required to understand hypervisor code. Operating systems, hardware devices, CPU.

- Hypervisor Threat Model.

Generalized hypervisor architecture. Attack surfaces, attack vectors, offensive research trends. Implementation-agnostic models.

- Logic bugs in hypervisors.

Case studies, where to find them, how to exploit.

Practicals:

- Developing an advanced hypervisor exploit, stage 0: vulnerability analysis (heap overflow), evaluation of exploitability, solving reachability.

DAY 2. SIMPLE OVERFLOW BUGS & OPEN SOURCE CODE

Lectures:

- Recap of memory safety concepts.

Classes of bugs, general considerations of exploitability by bug class, hypervisor specifics.

- Analyzing open source code.

Familiarizing with large code bases, reverse-engineering target architecture, finding bugs.

- Case studies.

Deep technical details of memory safety issues in various hypervisor implementations.

- Exploitation techniques.

Specific technical algorithms and primitives that work for memory corruption bugs in hypervisors.

Practicals:

- Developing an advanced hypervisor exploit, stage 1: create simple proof-of-concept codes for the target vulnerability.

DAY 3. NON-TRIVIAL BUGS & BINARY CODE

Lectures:

- Working with binary code.

Advanced-level recap of effective reverse-engineering techniques that are helpful when attacking proprietary hypervisors.

- Non-trivial memory safety bugs.

Uninitialized variables, race conditions, TOCTOU.

- Case studies.

Deep technical details of non-trivial security bugs in various hypervisor implementations.

- Exploitation techniques.

Specific technical algorithms and primitives that work for non-trivial bugs in hypervisors.

Practicals:

- Stage 2: adjusting proof-of-concepts to proprietary code of the target.

DAY 4. STATE OF THE ART

Lectures:

- Future of hypervisors.

Rustlang security essentials. Overview of hypervisors written in Rust. Hardware assist technology updates. Technological trends.

- Speculative execution bugs & Rowhammer.

Vs. the Hypervisor Threat Model. Known bugs that affect hypervisors. State of the art.

- Non-trivial exploits.

Principles of exploiting hard bugs & creating non-templated exploits.

Practicals:

- Stage 3: RCE PoC exploit & shellcode. Evaluating further work.

Note: the program may be changed.

TRAINING PACKAGES & FEES

This training is available live: online and in-person.

Online training is based on a modern streaming platform with high-quality audio and video, and a group chat. The instructor will be available for questions, feedback and technical support only for students with Advanced package. Lab setup is DIY.

Self-paced training offers exactly the same learning experience as online training, less the coaching factor (you'll have to time yourself) and near-instant availability of the instructor's feedback and answers.

All training packages were specifically optimized for online teaching.

LIVE ONLINE TRAINING

Basic package

What's included:
- Access to public online training.
- Instructor's feedback by email.
- Training slides and supplementary materials.
- Training completion certificate (upon request).

Price: €3,900.- per person.

Advanced package

What's included:
- Everything in the Basic package.
- Personal feedback and technical support from the instructor while in the class.
- Possibility to get a distinction certificate by demonstrating abilities and successful apprehension in the class.

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 (upon request).

Price: -

Advanced 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

Private and customized training may be available for groups of at least 10 persons.

PUBLIC TRAINING DATES & BOOKING

Refer to the schedule for the dates of the nearest public training.

This training is available exclusively for direct customer bookings through our website zerodayengineering.com and email. We may refuse to accommodate unauthorized third party bookings.

Bookings of public training seats are accepted via website checkout system. You may contact us directly for customized and private training bookings, direct bank payments and alternative payment methods.

WHAT TO EXPECT?

Quoted below are anonymized extracts of private feedback from attendees 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**."

Selected public (named) reviews from our students can be found on our website.

FURTHER INQUIRIES

E-mail: contact@zerodayengineering.com
Note: we typically respond to finalized booking and purchase communications within 1-2 business days. If you didn't receive a response, please check your spam folder first. Then send us a direct message.

RESERVED FOR NOTES

CHANGE LOG

August 24th, 2022: initial release of the document; training program v0.1.