Bug Hunting in Hypervisors

Training Description

BY REVERSE TACTICS

 contact@reversetactics.com

 https://www.reversetactics.com/

 https://x.com/Reverse_Tactics

 https://www.linkedin.com/company/reverse-tactics/

ABSTRACT

Hypervisors are complex software that play a critical role in modern infrastructure, but like any software, they're not immune to flaws which can be exploited by sophisticated attackers. This training dives into the technical depths of virtualization technologies and explores the flaws leading to virtual machine (VM) escapes. During this training, you will be able to sharpen your skills on multiple platforms from the initial analysis of a target to exploiting real world vulnerabilities.

The course explores the attack surfaces hypervisors expose to their guests, both statically and dynamically. By breaking down how virtual machines communicate with hypervisors and their internal components, participants will learn to apply their existing vulnerability research and exploitation skills to any virtualization software. The training also provides detailed insights for each studied target, including their architectures, typical vulnerabilities, and guidance for effective bug hunting.

This course is ideal for security researchers and vulnerability analysts who are already familiar with low-level systems programming and common exploitation techniques but are new to hypervisor internals. By the end of the training, participants will have a solid foundation in virtualization attack surfaces and vulnerability research as well as the ability to craft proof-of-concept exploits targeting hypervisors.

The course is designed to be given in 4 days of 7 hours.

Topics Covered

01.

Understanding hypervisor internals, components and architectures.

02.

Tools and techniques to effectively perform bug hunting on virtualization software.

03.

Methodology for navigating hypervisors code base, both open and closed source.

04.

Analyze and practice with vulnerabilities in QEMU/KVM, VirtualBox, VMware Workstation/ESXi.

PREREQUISITES AND REQUIREMENTS

Knowledge Prerequisites

- Basic programming skills in C and Python.
- Familiarity with low-level computer behaviour:
  - Userland vs Kernel execution.
  - Basic x86 processor architecture.
- Knowledge of reverse-engineering concepts and techniques.
- Understanding and experience of common C vulnerabilities and exploitation techniques:
  - Buffer overflows, use-after-free (UAF), race conditions, uninitialized variables.
  - ROP, heap massaging, ASLR bypass...

Technical Requirements

- A computer capable of running VMware Workstation Pro. It is free and downloadable from Broadcom website.
  - We are going to use nested virtualization with multiple different hypervisors which will work only on VMWare Workstation.
  - The processor of the computer must be an Intel or AMD x86 supporting VT-X or AMD-V. iMac based on ARM chips will not work.
  - Linux host preferred. If host is Windows, Hyper-V must be disabled during the training.
  - Trainee must have administrator privileges on its computer.
- HexRays IDA with x64 decompiler.
  - The IDA Free version is enough.
  - IDA Pro with ARM decompiler and scripting capabilities is preferred.

COVERED SUBJECTS

Hypervisor basics

- The definition and purpose of a hypervisor.
- Core architecture and components.
- x86 hardware-assisted virtualization:
  - VT-x/AMD-V.
  - EPT/SLAT.
- The necessity of device emulation and para-virtualization in providing hardware to the guest.

Interacting with the hypervisor

- Mechanisms for triggering guest-host interactions via MMIO, PMIO, and DMA.
- Using PCI/PCIe interfaces to communicate with specific emulated or para-virtualized devices.
- Tools and techniques for scripting guest-hypervisor communications.

Navigating and understanding the code base

- Exploration of the architectural layouts of QEMU/KVM, VirtualBox, VMware Workstation, and ESXi.
- Techniques for pinpointing areas of interest, such as memory mapping functions, device initialization, and handlers.
- Leveraging reverse engineering tools and methods to analyse complex, closed-source code.
- Reviewing strategies for locating documentation and resources to help symbolize closed-source code and understand internals.

Bug Hunting

- Identifying common attack surfaces.
- Recognizing bug types specific to virtualization.
- Tools and strategies for debugging hypervisors.
- Exploring fuzzing challenges and possible solutions.
- Rediscovering and exploiting n-day vulnerabilities as practical training for real-world bug hunting.

ASSIGNMENTS

Explore Device Emulation on QEMU/KVM

ASSIGNMENT 1

In this assignment, participants will explore the details of QEMU's device emulation to uncover potential vulnerabilities. The focus is on understanding and interacting with the hypervisor's behaviour through the guest system and analysing how I/O operations are managed.

Along the day, participants will explore common communication patterns and device interactions, and develop the skills needed to pinpoint their first vulnerabilities in a crafted emulated device.
In the final stage of this assignment, students will extend their knowledge to identify and trigger a real-world vulnerability that affected a previous version of QEMU.

VirtualBox Code Navigation and Exploit Development

ASSIGNMENT 2

This assignment introduces VirtualBox as a target for exploitation. Participants will explore aspects of VirtualBox's I/O handling and device emulation to identify vulnerabilities. Throughout the day, participants will work with VirtualBox's codebase, learning how to systematically navigate and analyse the architecture of an open-source hypervisor.

By applying learned methodologies, they will analyse memory mapping operations, locate potential bugs, and develop a proof-of-concept exploit for a selected vulnerability. The focus is on understanding typical bugs in hypervisors and how to approach them systematically.

Reverse & Bug Hunting in VMware

ASSIGNMENT 3

In the first part of the assignment, participants will reverse engineer components of VMware's closed-source hypervisors. They will map critical functions related to memory management and I/O handling. The assignment aims to provide insights into finding vulnerabilities in a closed-source environment, teaching participants to map code paths and identify areas prone to bugs or exploitation. Students will receive pre-symbolized IDA databases to assist in navigating the code.

The last part of the assignment brings together all skills developed during the training. Participants will analyse both VMware ESXi and Workstation to identify n-day vulnerabilities and attempt to develop proof-of-concept exploits. This exercise involves understanding the architectural differences between ESXi and Workstation, identifying attack surfaces, and crafting targeted exploits.

TRAINERS

Corentin BAYET

Corentin Bayet is the CTO of REverse Tactics and a seasoned security researcher with over 7 years of experience in vulnerability research and exploitation. His expertise lies in low-level technologies, including operating systems, kernels, and hypervisors.

Corentin has publicly demonstrated multiple VM escapes at high-profile events like Pwn2Own (2020, 2024), showcasing his advanced skills in hypervisor security.

He has also delivered impactful talks on bug hunting in virtualization at renowned conferences such as EkoParty 2020, GreHack 2023, and GreHack 2024.

Bruno PUJOS

Bruno Pujos is the CEO and founder of REverse Tactics, bringing over 10 years of experience as a security researcher specializing in low-level systems and virtualization technologies.

He has publicly demonstrated his expertise by achieving multiple VM escapes and privilege escalations on Windows at Pwn2Own (2020, 2022, 2024) .

Bruno is also an experienced trainer, having delivered advanced courses on reverse-engineering and bug hunting, including sessions focused on firmware and UEFI BIOS reverse engineering.