Module 1: Fuzzing Essentials with winAFL

Description

The first day of training introduces foundational fuzzing techniques with a focus on using **winAFL** on the library. Participants will delve into core fuzzing concepts, including effective **corpus generation** and advanced techniques to enhance the **fuzzing harness**. Hands-on exercises will guide participants in creating a basic harness that can handle various archive formats. A practical case study on **code execution vulnerabilities in WinRAR** illustrates real-world applications of fuzzing within Windows environments.

Main Targets Applications

- **LibArchive**: Open-source C library for reading and writing streaming archives.
- **WinRAR**: Widely-used file archiver for Windows, providing case-study insights.

Key Topics

- **winAFL**: Windows-based fuzzer for testing applications for vulnerabilities.
- **Fuzzing Fundamentals**: Concepts, corpus generation, and techniques for optimizing fuzzing outcomes.
- **Fuzzing Internals**: Deep dive into how fuzzing tools interact with binaries and libraries on Windows.

Learning Objectives

- Understand essential Windows internals relevant to fuzzing.
- Learn introductory fuzzing techniques.
- Develop skills to create a well-structured fuzzing corpus.

Module 2: Vulnerability discovery, Fuzzing Improvement and Coverage analysis using Jackalope

Description

On the second day, the training focuses on **fuzzing IrfanView** by utilizing tools such as **winAFL, Jackalope, and Lighthouse** to improve the analysis and triaging process. Participants will learn essential **triaging techniques**, conduct **coverage analysis**, and apply **debugging** strategies to uncover vulnerabilities. In a practical hands-on lab, participants will rediscover a **remote code execution (RCE) vulnerability** in **PSP files** and expand their fuzzing skills by working with **WEBP formats**. A comprehensive **ZDi report on fuzzing IrfanView** serves as a real-world reference, providing insights into professional vulnerability research practices.

Main Target Application

- **IrfanView**: An image viewer application that provides a learning case for vulnerability research and analysis.

Key Topics

- **winAFL**: Fuzzing tool tailored for Windows applications.
- **Jackalope**: Fuzzing tool tailored for Windows/linux/macos applications.
- **Lighthouse**: Code coverage visualizer for analyzing fuzzing effectiveness.

Learning Objectives

- Techniques for **triaging** vulnerabilities found during fuzzing.

- Gain skills in **coverage analysis** to assess fuzzing completeness.
- Develop effective **debugging** practices for vulnerability investigation.

Module 3: Structural Fuzzing and Symbol-less Reversing on PDF Applications

Description

Day three emphasizes **grammar** technique, focusing on applications that handle PDF files, such as **PDF-XChange** and **IrfanView's PDF plugin**. Participants will explore **fuzzing methodologies** for complex file structures and gain skills in **reversing binaries without symbols**—a critical technique in real-world vulnerability research. Key resources include the latest industry reports and an ICSE research paper that contextualizes these fuzzing techniques within modern security research.

Main Targets Applications

- **IrfanView PDF Plugin**: An additional target for handling PDF file fuzzing within IrfanView.
- **PDF-XChange**: Popular PDF viewing and editing software.

Key Topics

- **Jackalope**: Tool fuzzing can be used for grammar fuzzing.
- **Grammar**: Techniques for fuzzing complex file formats.
- **Reversing without Symbols**: Techniques for analyzing binary files without debugging symbols.

Learning Objectives

- Develop skills in **grammar-based fuzzing** for structured files like PDFs.
- Learn strategies for **reversing and analyzing** software binaries without the aid of symbols.

Module 4: Snapshot Fuzzing

Description

Day four shifts focus to **snapshot-based fuzzing** techniques using **video games** as a testing ground. The day's primary target, **Assault Cube**, provides a practical example for participants to apply snapshot fuzzing concepts with **Wtf** frameworks. Real-world case studies, including vulnerabilities in Assault Cube's map parser.

Main Target Application

- **Assault Cube**: Open-source, networked first-person shooter game with a focus on map parsing.

Key Topics

- **Snapshot Fuzzing**: Techniques for creating and analyzing snapshot-based fuzzing cases.
- **Wtf**: Snapshot fuzzing tools.

Learning Objective

- Understand **snapshot fuzzing** techniques for efficiently testing complex applications.

Who Should Attend?

This training is tailored for cybersecurity professionals, researchers, and engineers interested in mastering fuzzing techniques for vulnerability discovery across a variety of applications.

What You'll Gain

By the end of the training, participants will:

- Be proficient in multiple fuzzing techniques.
- Gain hands-on experience with state-of-the-art tools like winAFL, Jackalope, and Wtf.
- Learn practical skills to identify and triage real-world vulnerabilities effectively.

Prerequisite Knowledge

- Basic to intermediate proficiency in **C/C++**
- Familiarity with Windows internals and debugging tools
- Reverse Engineering Fundamentals

Required Hardware and Software

- Virtualization capable CPU(s)
- Minimum 8GB of RAM (for running one guest VM)
- Minimum 80 GB free disk space
- Host CPU intel
- Debugging Tools for Windows (Ida Pro, WinDBG) and Decompiler recommended.
- Virtualization Software (VMWare, VirtualBox)
- System Administrator access required on both host and guest OSs