Training:

Exploiting with Precision: Windbg Debugging Essentials for Security Professionals

Date of the training: **June 23rd – 24th, 2025** in Heidelberg, Germany

Book now using the voucher code: **TR25_HMTRAINING** and save an additional 5% of the current valid rate of any package!

Overview

There is nothing easy about using a debugger – whatever the debugger is. But it is so powerful when we know how to use it. Find a bug in an application? Analyse automatically critical operations? View all operations on a file or application network? Proof of concept to exploit a vulnerability? Each of these operations is matter of minutes, with a debugger. There is no need to guess what might happen in a specific situation. There is no need to write complex scripts or programs to test an application. The debugger makes life much easier for the security professionals who use it.

The aim of this workshop is twofold. On the one hand, to learn how to use Windows' flagship debugger: Windbg. The aim is to give to anyone (even beginners) a relatively quick and efficient experience to use Windbg. We learn the minimum of assembly language necessary to understand, when necessary, what the debugged program does. But the most important thing is practice. We see different exercises and practical cases, allowing anyone to do it alone with ease. On the other hand, always with a practical approach, we analyse an application with the aim of taking full control of it. The application is written specifically for this training course, and resembles several vulnerabilities based on real-life cases. It has a Remote Procedure Call (RPC) interface as well as a driver. The analysis of the RPC interface is carried out in such a way as to enable the participant to independently trace the analysis of such an interface in other software. The driver is fully exploited, enabling the participant to take full control of the system.

The workshop does not require any specific skill in reverse engineering. Indeed, the vulnerable application is provided with its source code, supposing the reverse engineering work has already been performed. The analysis context is source-code auditing and proof-of-concept exploitation. The workshop is provided with a dedicated application/service/driver set that illustrates various ways to exploit vulnerabilities. This training set is design with a Lab where we setup is done with the participants.

The goal is to understand how an attack with user access get access to "NT AUTHORITY\SYSTEM". To do this, we will go step by step. Firstly, we observe how the application interfaces with the service, then we take the control of the service, and finally we take control of the driver via the service. The elevation of privileges is then progressive (from the service to the driver). That way, each participant builds up his or her knowledge the way they progress. Each time, the participants pass different levels of security, exploiting conceptual, architectural or implementation issues.

The first day covers the most important commands that are useful in every day's life of a security professional, including the setup of an analysis environment (allowing to reverse anything in Windows) and tooling configuration. We will look at the most useful commands and efficient scripting practices with Windbg to boost our efficiency. Also, we will cover advanced techniques, particularly useful in the context of vulnerability research, such as time-travel-debugging, anti-debugging bypasses, and on the fly code execution in the target process from the debugger.

The second day focusses on the exploitation of the set of services and drivers, step-by-step, surpassing different kind of securities barriers on different levels (software / system). The goal is to identify vulnerabilities and to use many techniques with Windbg to calibrate different kind of exploits. That way, it will be possible for participants practice the different kind of attacks explained during the workshop.

Details of the Workshop

Day 1:

- Introduction to Windbg
    - General functioning of a debugger and capabilities
    - User-mode / Kernel-mode debugging
    - Live debugging, crash dump
    - Configuring Windbg (layout, symbols, GUI, modules)
    - Configuring the system for debugging (AEDebug, GFlag)
- Configuring the Lab
    - Setup the Virtual Machine to be kernel debugged (Network, USB, Baud rate)
    - Advanced kernel debug options (bcdedit)
    - Setup Windbg to debug the Virtual Machine
    - Debugger commands for kernel-mode debugging
- Minimal assembly (32-bit & 64-bit) understanding
    - Registers & memory
    - Basic operations
    - Conditional operations
    - Calling conventions
- User-mode debugging
    - Basic commands
        - Disassembly (and short assembly reminder)
        - Memory management
        - Step by step execution
        - Thread management
        - Call stack and frames
        - Breakpoints (code, data, conditional)
        - Data display and structure representation
    - Debugging user-mode process from kernel-mode
    - Advanced debugging
        - Time Travel Debugging
        - Bypassing anti-debugging
        - Direct code execution through debugging
- Vulnerability research & methodology
    - General methodology
    - Fuzzing, approaches and limitations
    - Source code audit
    - Application/Driver verifier

Day 2:

- “Minimal” Windows Internals
    - Process, thread, driver, service
    - Memory management (allocation, rights, stack, heap)
    - Object system (handle, security, ACL/ACE, Privileges)
    - Inter Process communication through RPC
    - Windows API documentation
- Lab introduction
    - Introduction to the vulnerable service
    - Attack surface exploration
        - Static analysis vs dynamic analysis
        - Observing the target with Windbg / Sysinternals’ tooling
    - Interacting with the vulnerable service
    - Debugging an auto-boot service
    - Exploring specific features in the vulnerable service
- Lab: Vulnerability exploitation (user-mode)
    - Address leak
    - Read-what-where
    - Write-what-where
    - Dll hijacking
    - Remote Code execution with shellcode
    - Elevation of privileges and full exploit chain creation
- Lab: Vulnerability exploitation (kernel-mode)
    - Understanding service/driver communication
    - Hijacking the communication
    - Read-what-where
    - Write-what-where
    - Elevation of privileges from kernel to user-mode, exploit chain creation

Who should attend this training?

- Vulnerability researchers
- Red Team Members
- Pentesters
- Reverse Engineers
- Students

Requirements

The attendees should have:

- Basic knowledge of programming. A script or a compiled language is a required.
- Basic knowledge of reverse engineering. Knowing x86 and x64 is definitively a plus without being a must.

Hardware/software requirements:

- A laptop with Intel/AMD CPU (x86 or x64) and administrator rights
- At least 100 GB of free disk space
- At least 16 GB of RAM
- Possibility to install applications from the Windows Store
- Hyper-V or Virtual Box with a clean Windows 10 or 11 image already installed

About the Trainers:

Dr. Baptiste David is an IT security specialist at ERNW, specialized in Windows operating system. His research is mainly focused on malware analysis, reverse engineering, security of the Windows operating system platform, kernel development and vulnerabilities research. He has given special courses and trainings in different universities in Europe. Also, he gives regularly talks on different conferences including Black Hat USA, Defcon, Troopers, Zero Night, Cocon, EICAR, ECCWS…

Booking

Recommended online booking of trainings through:
https://troopers.de/tickets/

Voucher code: TR25_HMTRAINING
Using this voucher code automatically gives you an additional 5% off the current valid price! You can register with this code until seats have run out.

Contact

Need assistance? Don't hesitate to call us. We are fluent in English and German. +49 6221 480390 or info@troopers.de

Booking is also possible offline through your trusted partner:

HM Training Solutions, Falkenstrasse 6, 63820 Elsenfeld, Germany
+49 6022 508200 / info@hm-ts.de
+49 6022 5089999 / https://hm-ts.de