Welcome to the Windows Kernel Rootkit Techniques Training

Designed and presented by:

CodeMachine

codemachine.com
@codemachineinc

Introduction

- T Roy
  - 25 years experience
  - Author, instructor, consultant
    - Focus on Windows security
  - Founder and President of CodeMachine
- CodeMachine
  - Security Research and Training company
    - Based in USA. Founded in 2001
  - Expertise
    - Reverse engineering, offensive and defensive development
    - Endpoint security, kernel drivers, forensic tools etc.
    - Design review, debugging, memory dump analysis
  - Online and onsite instructor led training courses
- Contact information codemachine@outlook.com

Windows Security Training Courses

WININT — Windows Internal Architecture

Security focused behind the scenes walkthrough of Windows internals pertaining to user mode components, applications and services.

WINMAL — Windows Malware Techniques

Post-exploitation techniques used at different execution stages by native Windows PE-file based malware and user-mode implants.

KERINT — Windows Kernel Internals

Deep dive into the internals of the Windows kernel from a security perspective with an emphasis on algorithms, data structures, and kernel debugger usage.

KERDEV — Windows Kernel Development

Windows kernel software driver development essentials with emphasis on kernel functionality pertinent to offensive and defensive security software.

KERRKT — Windows Kernel Rootkits

End-to-end view of the modus-operandi of kernel mode rootkits and how security mitigations in modern versions of Windows attempts to thwart them.

KERDBG — Windows Kernel Debugging

Techniques for triaging, fault isolation, root cause analysis of kernel memory dumps from system crashes and hangs caused by kernel-mode drivers.

Daily Schedule

- Class Schedule (all times in US PST)
  - 0900 hours start
  - 1030 hours morning break (15 minutes)
  - 1215 hours lunch break (60 minutes)
  - 1445 hours early afternoon break (15 minutes)
  - 1630 hours late afternoon break (15 minutes)
  - 1800 hours finish
- Each module in this training course comprises of
  - Theory, demos, hands-on labs (some for homework), code walkthroughs, summary and quiz (optional)
- Technically intensive course
  - Make yourself comfortable, relax and enjoy the class

If you have a question you **must** ask it.

Logistics

- Asking questions
  - Questions always welcome
  - Ask anytime using the GotoTraining chat window
- Debugger logs
  - All debugger logs will be made available to you daily
  - Includes all commands and debugger output
- Virtual training
  - Lag between audio and screen updates
  - Intermittent audio quality degradation or drops
  - Accidental muting
- Training Feedback
  - Daily feedback https://forms.office.com/r/DvmrWPZ34u
  - Completion feedback https://forms.office.com/r/qSy9QWsy60

Course Material Overview

- Class notes and file shares
  - Files will be shared through OneDrive
  - Class notes updated in real-time shared through Google Docs
- Training downloads
  - Directory structure and content are described in the document *TrainingDownloadContents.pdf* on OneDrive
    - host.zip -> c:\cm
    - guest.zip -> \\winlabvm\pub
    - symbols.zip -> \\winlabvm\pub\sym
- System Setup
  - Enterprise WDK (EWDK)
  - WinDBG Preview (live kernel debugging)
  - Virtual Machine (start in debug mode)
  - Break into VM using WinDBG (Alt+Del)
  - Symbol path (.sympath)
  - Memory Dumps

Hands-on Labs

- Download *KERRKT_labs.zip* from OneDrive
  - Unzip the contents into *c:\CM\src*
  - The *labs* folder contains the source code templates for your programming exercises
  - The *src* folder contains of source code of fully functional applications and kernel modules
- Lab solutions will be provided at the end of the class
- Download *KERRKT_LabExercises.pdf* from OneDrive
  - List of all the hands-on lab exercises which we will be working on throughout this class
  - Detailed instructions for all the hands-on lab exercises
- Download *BuildingWithEWDK.pdf* from OneDrive
  - Contains the steps that you must follow to code, build, stage, copy, install, and test applications/kernel modules