



WINDOWS MALWARE ANALYSIS AND MEMORY FORENSICS

Expert-Led Cybersecurity Training · Beginner to Advanced

COURSE OVERVIEW

Malware analysis and memory forensics are powerful analysis and investigative techniques used in reverse engineering, digital forensics, and incident response. With adversaries getting sophisticated and carrying out advanced malware attacks on critical infrastructures, Data Centers, private and public organizations, it is essential for cyber-security professionals to have the necessary skills to detect, respond and investigate such intrusions. Malware analysis and memory Forensics have become a must-have skill for fighting advanced malwares, targeted attacks, and security breaches. This hands-on training teaches the concepts, tools, and techniques to analyze, investigate, and hunt malwares by combining two powerful techniques malware analysis and memory forensics. After taking this course, attendees will be better equipped with the skills to analyze, investigate, and respond to malware-related incidents.

This course will introduce attendees to the basics of malware analysis, reverse engineering, Windows internals, and memory forensics and then it gradually progresses deep into more advanced concepts of malware analysis & memory forensics. Attendees will learn to perform static, dynamic, code, and memory analysis. To keep the training completely practical, it consists of various scenario-based hands-on labs after each module which involves analyzing real-world malware samples and investigating malware infected memory images (crimewares, APT malwares, Fileless malwares, Rootkits, etc).

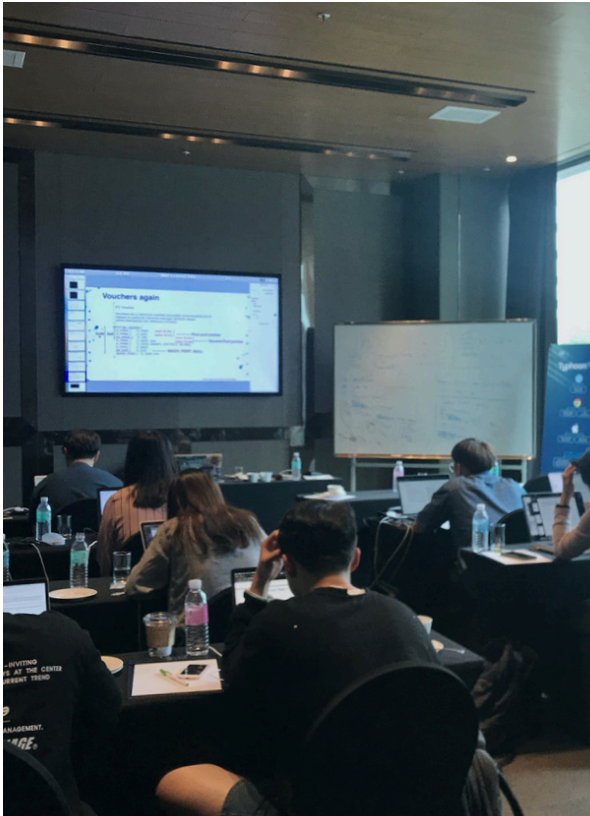
This hands-on training is designed to help attendees gain a better understanding of the subject in a short span of time. Throughout the course, the attendees will learn the latest techniques used by the adversaries to compromise and persist on the system. In addition to that, it also covers various code injection, hooking, and rootkit techniques used by adversaries to bypass forensic tools and security products. In this training, you will also gain an understanding of how to integrate malware analysis and memory forensics techniques into a custom sandbox to automate the analysis of malicious code.

KEY LEARNING OBJECTIVES

- How malware and Windows internals work
- How to create a safe and isolated lab environment for malware analysis
- Tools and techniques to perform malware analysis
- How to perform static analysis to determine the metadata associated with malware
- How to perform dynamic analysis of the malware to determine its interaction with process, file system, registry, and network
- How to perform code analysis to determine the malware functionality
- How to debug malware using tools like IDA Pro and x64dbg
- How to analyze downloaders, droppers, keyloggers, fileless malwares, HTTP backdoors, etc.
- Understanding various persistence techniques used by the attackers
- Understanding different code injection techniques used to bypass security products
- What is Memory Forensics and its use in malware and digital investigation
- Ability to acquire a memory image from suspect/infected systems
- How to use open source advanced memory forensics framework (Volatility)
- Understanding of the techniques used by the malwares to hide from Live forensic tools
- Understanding of the techniques used by Rootkits (code injection, hooking, etc.)
- Investigative steps for detecting stealth and advanced malware
- How memory forensics helps in malware analysis and reverse engineering
- How to incorporate malware analysis and memory forensics in the sandbox
- How to determine the network and host-based indicators (IOC)
- Techniques to hunt malwares

WHY SHOULD YOU TAKE THIS COURSE?

This is a completely hands-on course designed for beginners and intermediate students. Instead of just slides, attendees will get a chance to exploit all of the vulnerabilities taught by the instructors. Slack/Discord channel is created before the course for the students so that they can be adequately prepped in terms of hardware and software before the class.



WHO SHOULD ATTEND?

This course is intended for

- Forensic practitioners, incident responders, cyber-security investigators, security researchers, malware analysts, system administrators, software developers, students, and curious security professionals who would like to expand their skills
- Anyone interested in learning malware analysis and memory forensics.

PREREQUISITE KNOWLEDGE

- Working knowledge of cybersecurity and pentesting fundamentals
- Basic Windows skills and command-line proficiency
- Understanding of fundamental programming concepts and looping structures in at-least one higher-level language
- Basic Windows binary assembly knowledge is recommended, but not required
- Working knowledge of malware analysis concepts is recommended, but not required

WHAT WILL THE STUDENTS GET?

- Certificate of completion for the Training program
- Sample Malwares used during the class
- Slack access for the class and after for regular mobile security discussions

HARDWARE/SOFTWARE REQUIREMENT

- Laptop with a minimum of 6GB RAM and 40GB free hard disk space
- VMware Workstation Pro or VMware Fusion Pro (now free for personal use)
- Windows Operating system (preferably Windows 10 64-bit, even Windows 8 and lower versions are fine) installed inside VMware Workstation Pro/Fusion Pro. You must have full administrator access to the Windows operating system installed inside VMware Workstation Pro/Fusion Pro.

Note: VirtualBox is not suitable for this training. The lab setup guide will be sent a few weeks prior to the class.

Module 1: Introduction to Malware Analysis

- What is Malware
- What they do
- Why malware analysis
- Types of malware analysis
- Setting up an isolated lab environment

Module 2: Static Analysis

- Fingerprinting the malware
- Extracting strings
- Determining File obfuscation
- Pattern matching using YARA
- Fuzzing hashing & comparison
- Understanding PE File characteristics
- Disassembly
- MalDoc analysis: VBA/XLM macros, RTF, OneNote, HTML smuggling, ISO/LNK containers
- Hands-on lab exercise involves analyzing a real malware sample

Module 3: Dynamic Analysis/Behavioural analysis

- Dynamic Analysis Steps
- Understanding Dynamic Analysis tools
- Simulating services
- Performing Dynamic Analysis
- Monitoring process, filesystem, registry, and network activity
- Determining the Indicators of compromise (host and network indicators)
- Hands-on lab exercise involves analyzing a real malware sample
- Automating Malware Analysis(sandbox)
- Custom Sandbox Overview
- Working of Sandbox
- Sandbox Features
- Demo Analyzing malware in the custom sandbox
- Living-off-the-Land Binaries (LOLBins/LOLBAS)

Module 4: Malware Persistence Methods

- Run registry key
- Scheduled Tasks
- Startup Folder
- Service
- Winlogon registry entries
- Image File Execution Options (IFE0)
- Accessibility programs
- Applnit_DLLs
- DLL Search order hijacking
- COM Hijacking
- Browser-hijacker persistence: ChromeLoader case study (scheduled task + malicious browser extension)
- Hands-on lab exercise involves analyzing a real malware sample

Module 5: Code Analysis

- Code Analysis Overview
- Disassembler & Debuggers
- Code Analysis Tools
- Basics of IDA Pro
- Basics of x64dbg (with Snowman/HEX-RAYS decompiler)
- Understanding the API calls
- Reversing Malware functionalities (Downloader, dropper, keylogger, code injection, HTTP backdoor)
- Hands-on lab exercise involves analyzing a real malware sample

Module 6: Introduction to Memory Forensics

- What is Memory Forensics
- Why Memory Forensics
- Steps in Memory Forensics
- Memory acquisition and tools
- Acquiring memory From a physical machine
- Acquiring memory from the virtual machine
- The hands-on exercise involves acquiring the memory

Module 7: Volatility 3 Overview

- Introduction to Volatility 3 Advanced Memory Forensics Framework
- Volatility 3 Installation
- Volatility 3 basic commands
- Automatic symbol resolution and ISF symbol tables
- Volatility 3 help options
- Running the plugin

Module 8: Investigating Process

- Understanding Process Internals
- Process(EPROCESS) Structure
- Process organization
- Process Enumeration by walking the double linked list
- process relationship (parentchild relationship)
- Understanding DKOM attacks
- Process Enumeration using pool tag scanning
- Volatility plugins to enumerate processes
- Identifying malware process
- Hands-on lab exercise(scenariobased) involves investigating malware infected memory

Module 9: Investigating Process handles & Registry

- Objects and handles overview
- Enumerating process handles using Volatility
- Understanding Mutex
- Detecting malware presence using a mutex
- Understanding the Registry
- Investigating common registry keys using Volatility
- Detecting malware persistence
- Hands-on lab exercise(scenariobased) involves investigating malware infected memory



Module 10: Investigating Network Activities

- Understanding malware network activities
- Volatility Network Plugins
- Investigating Network connections
- Investigating Sockets
- Hands-on lab exercise(scenariobased) involves investigating malware infected memory

Module 11: Investigation Process Memory

- Process memory Internals
- Listing DLLs using Volatility
- Identifying hidden DLLs
- Dumping malicious executable from memory
- Dumping DLL's from memory
- Scanning the memory for patterns(yarascan)
- Case study: extracting Cobalt Strike beacon configurations from memory
- Case study: identifying reflectively-loaded modules (Sliver, Brute Ratel, Havoc) in process memory
- Hands-on lab exercise(scenariobased) involves investigating malware infected memory

Module 12: Investigating UserMode Rootkits & Fileless Malwares

- Code Injection
- Types of Code injection
- Remote DLL injection
- Remote Code injection
- Reflective DLL injection
- Hollow process injection
- Process Doppelgänger & Process Ghosting
- Direct/Indirect syscall invocation (Hell's Gate, Halo's Gate, Tartarus' Gate)
- AMSI bypass techniques (patching AmsiScanBuffer, hardware breakpoints)
- ETW patching and unhooking userland EDR hooks
- Cobalt Strike beacon detection & memory artifact analysis
- Demo Case Study
- Hands-on lab exercise(scenariobased) involves investigating malware infected memory

Module 13: Memory Forensics in Sandbox technology

- Sandbox Overview
- Integrating Memory Forensics into a sandbox
- Demo showing the use of memory forensics in a custom sandbox

Module 14: Investigating KernelMode Rootkits

- Understanding Rootkits
- Understanding Functional call traversal in Windows
- Level of Hooking/Modification on Windows
- BYOVD (Bring Your Own Vulnerable Driver) attacks – case studies (e.g., RTCore64.sys, GIGABYTE driver, Lazarus' FudModule)
- Kernel callbacks tampering and PPL (Protected Process Light) bypass
- Detecting EDR kernel-component tampering in memory
- Kernel Volatility plugins
- Hands-on lab exercise(scenariobased) involves investigating malware infected memory
- Demo Rootkit Investigation

Module 15: Memory Forensic Case Studies

- Demo - Hunting an APT malware from Memory



About the company

8kSec is a foremost cyber security research company offering exceptional training and consulting services to aid clients in enhancing their security stance. Our experts possess extensive experience in delivering specialised cybersecurity training and consulting to several commercial and defence organisations across the United States, Europe, and the Middle East and North Africa region.

Get in touch

[8kSec.io](https://8ksec.io)

trainings@8ksec.io



The information in this document is subject to change without notice.