

Malware Analysis Crash Course

Think you are ready for a fire hose blasting the art of Malware Analysis into your brain? Our Malware Analysis Crash Course is just the class for you. In 2 days of intense instruction, students will be led on a fast paced journey that covers the art of malware analysis. If this class was a car, we would be saying prepare to go from “zero to sixty” in the blink of an eye.

Almost every Incident Response involves some Trojan, back door, virus component, or rootkit. Incident Responders must be able to perform rapid analysis on the malware encountered in an effort to determine the purpose of unknown code. This course provides a rapid introduction to the tools and methodologies used to perform dynamic and static analysis on portable executable programs found on Windows systems. Students will learn how to infer the functionality of a program by analyzing disassembly and by watching how it changes a system as it runs; how to extract investigative leads from host and network-based indicators associated with a malicious program; and how to identify specific coding constructs in disassembly. They will be taught the art of dynamic analysis, and they will be taught about several Windows APIs most often used by malware authors.

Each section is filled with in class demonstrations, exercises where the students follow along with the instructor, and labs where the students practice what they have learned on their own.

Duration	2 days
Who Should Attend	Information technology staff, information security staff, corporate investigators or others requiring an understanding of how malware works and the steps and processes involved in Malware Analysis.
Prerequisites	Excellent knowledge of computer and operating system fundamentals is required. Some exposure to software development is highly recommended.
Students Will Learn	<ul style="list-style-type: none">▪ Static Program Analysis without Disassembly▪ Dynamic Program Analysis without Debugging▪ Methodology differences between static and dynamic analysis▪ Bits, bytes, binary, decimal, hexadecimal and converting values between the various numbering conventions▪ How to perform dynamic analysis using system monitoring utilities to capture the system, registry and network activity generated during malware analysis▪ The fundamentals of assembly language programming▪ Recognizing coding constructs in disassembly▪ Debuggers▪ Windows Internals and APIs
Exercises / Labs	<ul style="list-style-type: none">▪ Dynamic and static analysis of unknown binaries▪ Recognizing Coding Constructs in Disassembly (multipart)▪ Debuggers▪ Understanding Windows internals (multipart): file API, loader, networking, registry usage, threads▪ Final lab
Course Materials	<ul style="list-style-type: none">▪ Student manual▪ Class handouts▪ MANDIANT gear
Suggested Next Courses	<ul style="list-style-type: none">▪ MANDIANT Intermediate Malware Analysis▪ MANDIANT Advanced Malware Analysis▪ MANDIANT Enterprise Incident Response
Contact	1.800.647.7020 education@mandiant.com www.mandiant.com/education.htm

Introduction to Malware

This section introduces the methodology behind malware analysis and establishes a baseline for advanced discussions.

Simple Malware Analysis Tools

Students will be introduced to a variety of “shortcuts” that can be used to facilitate the analysis of commonly used malware. Students will learn how to generate MD5 checksums to identify known malware, various websites which allow upload of suspicious code, and other approaches which can give a malware analyst a head start on the analysis process. Static analysis tools and techniques that do not use disassembly will be covered, as well as, dynamic analysis tools and techniques that do not use debugging.

Disassembly

In this section we dig deep into static analysis of the code that makes up a program using a disassembler to transfer the machine code to a more manageable representation. Topics include the following major topics:

- x86 assembly language
- Reversing basics: branches, loops and switches
- Reversing basics: functions
- Cross references
- Imports and Exports
- Type Libraries (Delphi, NTDDK, NTAPI, etc)
- A String is a String?: strings in ASCII, Unicode, Pascal, C, and Delphi
- Defining Arrays
- Defining Structures
- Standard Library Functions and FLIRT
- IDC Scripts, IDAPython, IDARub
- IDA Plugins

Debuggers

Although system monitoring tools provide a simple method for watching a program’s behavior as it runs, there are many times when an analyst will need to observe and monitor the internal workings of a running program rather than just watching the external behavior. Debuggers provide a means to observe and change both the code and data of a program as it runs. Specific topics discussed in this section will include:

- x86 Hardware Debugging Support
- Viewing/modifying memory, disassembly, registers, stack, call tree
- Labeling, commenting, bookmarks
- Breakpoints, conditional breakpoints, hardware breakpoints
- Controlling execution by stepping
- Running traces, back tracing
- Finding and modifying data of interest
- Patching binaries for temporary or permanent behavior modification
- Debugging a dll

Windows Internals

This course focuses on the analysis of malware on the Microsoft Windows “environment”. Topics discussed will include:

- PE file format
- Loader and dynamically linked libraries
- Windows API Overview
- Windows types
- Windows file system internals
- Registry usage
- Processes and threads
- Windows networking functions
- Windows Native API

Final Lab

Prepare to be challenged. We have taken malware collected by MANDIANT via their Incident Response team, crafted a new backdoor based on what we observed, and are turning the resultant executable over to the students for complete analysis. The lab has structured components, but the students are given free reign to uncover the mysteries of the malware.