

MALWARE ANALYSIS III – ADVANCED MALWARE ANALYSIS

Many malware authors take deliberate steps to thwart the reverse engineering of their tools. Students will learn to combat sophisticated malware head-on by studying its anti-analysis techniques. This course focuses on advanced topic areas related to combating malware defense mechanisms, and as such, a practiced and robust malware analysis skill set is required. Before learning specific malware anti-analysis techniques, students will arm themselves with critical skills by learning to script IDA Pro and various debuggers to overcome challenging or repetitive tasks. Students will learn detailed information about defeating packed and armored executables and be challenged to defeat several difficult specimens throughout the course. Malware stealth techniques such as process injection and rootkit technology will be introduced and tools and methodologies will be presented to aid analysis of such techniques. 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	5 Days
Who Should Attend	Information security staff, forensic investigators or others requiring an understanding of how to overcome difficult challenges in malware analysis.
Prerequisites	Training or experience in malware analysis and excellent knowledge of computer and operating system fundamentals is required. Some exposure to software development is highly recommended. Attendance in MANDIANT Malware II – Intermediate Malware Analysis, while not required, is extremely beneficial.
Students Will Learn	<ul style="list-style-type: none">▪ IDA Pro Scripting▪ Scriptable Debuggers▪ How to Conduct Analysis of Nontraditional Programs▪ How to Unpack Strongly Protected Binaries▪ How to Defeat Anti-Reverse Engineering Techniques▪ How to Recognize and Defeat Data Encryption and Encoding Techniques▪ How to Capture and Analyze Stealth Malware
Course Materials	<ul style="list-style-type: none">▪ Student manual▪ Class handouts▪ MANDIANT gear▪ Free Tools CD with course tools and scripts
Contact	1.800.647.7020 education@mandiant.com www.mandiant.com/education.htm

Advanced Static Analysis

Clever malware authors will attempt to make the reverse engineering process difficult by forcing the malware analyst to conduct fairly complex tasks in repetitive fashion. Defeating malware of this nature may take days or weeks if attempted by hand. In this section we will discuss scripting the IDA Pro disassembler to tackle these challenges as well as other tools to enhance and quicken the static analysis process.

Advanced Dynamic Analysis

Several scriptable debugging systems will be introduced in this section to arm the student with techniques to tackle the difficult protection mechanisms and packers used by the more devious malware today. The Windows Debugging API will also be introduced with in-depth coverage of debugging internals. Students will be taught to use tools to script and control the debugging process in C or Python as well as proprietary debugging languages. Topics in this section include the following:

- OllyDbg with OllyScript
- The PaiMai Framework
- Immunity Debugger
- Sample source code for using the Windows Debugging API

Other advanced Dynamic Analysis techniques such as API interception will be covered.

Nontraditional Binaries

The malware author's choice of programming language can play a large role in hindering the malware analysis process. Without detailed understanding of the inner-workings of compiled binaries from many modern high-level programming languages, understanding a program's high level constructs might remain an elusive target. In this section we will discuss and analyze binaries compiled with C++, .NET, Visual Basic, Delphi and Perl2Exe.

Packers

Approximately 70% of the malware that MANDIANT encounters in Incident Response engagements is packed. Packing of malware is a major issue for the analyst since any static analysis of packed code is almost entirely useless. In this section we discuss the internals of several commonly used packers, present general techniques that can be used to "unpack" code, and present the tools and methods necessary to "reconstitute" an unpacked binary into an easier to analyze unpacked version of the packed program.

Anti-Reverse Engineering

Malware analysis is a cat and mouse game in which both sides are constantly battling for the upper hand by developing new methods, tools, and techniques. The malware author's job is to develop software that can collect data, run undetected, and frustrate reverse-engineering efforts just enough to make it not worth the analyst's effort. As such, the malware analyst's job is to develop techniques that can quickly and reliably uncover the inner workings of malicious programs. In this section we discuss the latest techniques that malware authors are using to make analysis more difficult. Topics will be updated for each class, but will likely include:

- Virtual Machine detection
- Debugger Detection
- Hardware breakpoint detection
- Software breakpoint detection
- Anti-Disassembler tricks

Data Encryption and Encoding

An important "trick of the trade" in malware analysis is the ability to recognize and reverse various methods used to encode data. Malware authors encode data to help "hide" the data from a casual observer. For example, strings within a binary file or data in a network packet will be obfuscated so running the "strings" command or sniffing on the network will not immediately reveal the program's *evil* intentions.

Stealth Malware

A fascinating and increasingly relevant area of malware defense strategies is to employ stealth techniques to prevent the detection and capture of the malware itself. This section will cover in-depth the technique of process injection in all its forms. Several practical samples will be presented for detection and analysis. Kernel level Rootkit technologies will be introduced and tools will be introduced to aid in their analysis.

Analyzing Shellcode and Exploits

An agile malware analyst must learn to step outside the bounds of conventional software. Malicious code can come in many forms besides a standard executable or DLL file. This section will introduce the student to the challenges of reverse engineering software exploits and the malicious executable payloads known as shellcode.

Final Labs

The capstone of the course is challenging the student to apply all their new found knowledge and skill against a series of well protected malware samples in a real world scenario.