SANS The most trusted source for cybersecurity training, certifications, degrees, and research

GIAC CERTIFICATIONS

FOR610: Reverse-Engineering Malware: Malware Analysis Tools and Techniques

GREM
Reverse Engineering Malware
giac.org/grem

6	36	Laptop
Day Program	CPEs	Required

You Will Be Able To

- Build an isolated, controlled laboratory environment for analyzing code and behavior of malicious programs
- Employ network and system-monitoring tools to examine how malware interacts with the file system, registry, network, and other processes in a Windows environment
- Uncover and analyze malicious JavaScript and VBScript components of web pages, which are often used by exploit kits for drive-by attacks
- Control relevant aspects of the malicious program's behavior through network traffic interception and code patching to perform effective malware analysis
- Use a disassembler and a debugger to examine the inner workings of malicious Windows executables
- Bypass a variety of packers and other defensive mechanisms designed by malware authors to misdirect, confuse and otherwise slow down the analyst
- Recognize and understand common assembly-level patterns in malicious code, such as DLL injection and anti-analysis measures
- Assess the threat associated with malicious documents, such as PDF and Microsoft Office files
- Derive Indicators of Compromise (IOCs) from malicious executables to strengthen incident response and threat intelligence efforts

> **"This is a truly a step-by-step mentorship course. The content is immediately applicable to DFIR job roles."**
>
> — Chad Reams, **Parsons Inc.**

Learn to turn malware inside out! This popular reversing course explores malware analysis tools and techniques in depth. FOR610 training has helped forensic investigators, incident responders, security engineers, and IT administrators acquire the practical skills to examine malicious programs that target and infect Windows systems.

Understanding the capabilities of malware is critical to your ability to derive threat intelligence, respond to cybersecurity incidents, and fortify enterprise defenses. This course builds a strong foundation for reverse-engineering malicious software using a variety of system and network monitoring utilities, a disassembler, a debugger, and many other freely available tools.

The course begins by establishing the foundation for analyzing malware in a way that dramatically expands upon the findings of automated analysis tools. You will learn how to set up a flexible laboratory to examine the inner workings of malicious software, and how to use the lab to uncover characteristics of real-world malware samples. You will also learn how to redirect and intercept network traffic in the lab to explore the specimen's capabilities by interacting with the malicious program.

The course continues by discussing essential assembly language concepts relevant to reverse engineering. You will learn to examine malicious code with the help of a disassembler and a debugger in order to understand its key components and execution flow. In addition, you will learn to identify common malware characteristics by looking at suspicious Windows API patterns employed by malicious programs.

Next, you will dive into the world of malware that thrives in the web ecosystem, exploring methods for assessing suspicious websites and de-obfuscating malicious JavaScript to understand the nature of the attack. You will also learn how to analyze malicious Microsoft Office, RTF, and PDF files. Such documents act as a common infection vector as a part of mainstream and targeted attacks. You will also learn how to examine "file-less" malware and malicious PowerShell scripts.

Malware is often obfuscated to hinder analysis efforts, so the course will equip you with the skills to unpack executable files. You will learn how to dump such programs from memory with the help of a debugger and additional specialized tools, and how to rebuild the files' structure to bypass the packer's protection. You will also learn how to examine malware that exhibits rootkit functionality to conceal its presence on the system, employing code analysis and memory forensics approaches to examining these characteristics.

FOR610 malware analysis training also teaches how to handle malicious software that attempts to safeguard itself from analysis. You will learn how to recognize and bypass common self-defensive measures, including code injection, sandbox evasion, flow misdirection, and other measures.

The course culminates with a series of Capture-the-Flag challenges designed to reinforce the techniques learned in class and provide additional opportunities to learn practical, hands-on malware analysis skills in a fun setting.

Hands-on workshop exercises are a critical aspect of this course. They enable you to apply malware analysis techniques by examining malicious software in a controlled and systemic manner. When performing the exercises, you will study the supplied specimens' behavioral patterns and examine key portions of their code. To support these activities, you will receive pre-built Windows and Linux virtual machines that include tools for examining and interacting with malware.

Section Descriptions

SECTION 1: Malware Analysis Fundamentals

Section 1 lays the groundwork for malware analysis by presenting the key tools and techniques useful for examining malicious programs. You will learn how to save time by exploring Windows malware in several phases. Static properties analysis examines meta data and other file attributes to perform triage and determine the next course of action. Behavioral analysis focuses on the program's interactions with its environment, such as the registry, file system, and network. Code analysis focuses on the specimen's inner workings and makes use of debugging tools such as x64bg. You will learn how to set up and use a flexible laboratory to perform such an analysis in a controlled manner, becoming familiar with the supplied Windows and Linux (REMnux) virtual machines. You will then learn how to begin examininng malware in your lab-with guidance and explanations from the instructor to reinforce the concepts discussed throughout the day. The tools introduced in this section include PeStudio, Process Hacker, Process Monitor, x64dbg, API Monitor, and others.

TOPICS: Assembling a Toolkit for Effective Malware Analysis; Examining Static Properties of Suspicious Programs; Performing Behavioral Analysis of Malicious Windows Executables; Performing Static and Dynamic Code Analysis of Malicious Windows Executables; Interacting with Malware in a Lab to Derive Additional Behavioral Characteristics

SECTION 2: Reversing Malicious Code

Section two focuses on examining malicious Windows executables at the assembly level. You will discover approaches for studying inner workings of a specimen by looking at it through a disassembler and, at times, with the help of a debugger. The section begins with an overview of key code-reversing concepts and presents a primer on essential x86 Intel assembly concepts, such as instructions, function calls, variables, and jumps. You will also learn how to examine common assembly constructs, such as functions, loops, and conditional statements. The material will then build on this foundation and expand your understanding to incorporate 64-bit malware, given its growing popularity. Throughout the discussion, you will learn to recognize common characteristics at a code level, including HTTP command and control, keylogging, and command execution.

TOPICS: Understanding Core x86 Assembly Concepts to Perform Malicious Code Analysis; Identifying Key Assembly Logic Structures with a Disassembler; Following Program Control Flow to Understand Decision Points During Execution; Recognizing Common Malware Characteristics at the Windows API Level (Registry Manipulation, Keylogging, HTTP Communications, Droppers); Extending Assembly Knowledge to Include x64 Code Analysis

Who Should Attend

- Individuals who have dealt with incidents involving malware and want to learn how to understand key aspects of malicious programs
- Technologists who have informally experimented with aspects of malware analysis and are looking to formalize and expand their expertise in this area
- Forensic investigators and IT practitioners looking to expand their skill sets and learn how to play a pivotal role in the incident response process

SECTION 3: Malicious Web and Document Files

Section 3 focuses on examining malicious web pages and documents, which adversaries can use to directly perform malicious actions on the infected system and launch attacks that lead to the installation of malicious executable files. The section begins by discussing how to examine suspicious websites that might host client-side exploits. Next, you will learn how to deobfuscate malicious scripts with the help of script debuggers and interpreters, examine malicious Microsoft Office macros, and assess the threats associated with PDF and RTF files using several techniques. The tools introduced in this section include Fiddler, SpiderMonkey, pdf-parser.py, scdbg, and others.

TOPICS: Interacting with Malicious Websites to Assess the Nature of Their Threats; De-obfuscating Malicious JavaScript Using Debuggers and Interpreters; Analyzing Suspicious PDF Files; Examining Malicious Microsoft Office Documents, Including Files with Macros; Analyzing Malicious RTF Document Files

SECTION 4: In-Depth Malware Analysis

Section 4 builds on the approaches to behavioral and code analysis introduced earlier in the course, exploring techniques for uncovering additional aspects of the functionality of malicious programs. The section begins by discussing how to handle packed malware. We will examine ways to identify packers and strip away their protection with the help of a debugger and other utilities. We will also walk through the analysis of malware that employs multiple technologies to conceal its true nature, including the use of registry, obfuscated JavaScript and PowerShell scripts, and shellcode. Finally, we will learn how malware implements spyware and usermode rootkit functionality to perform code injection and API hooking, examining this functionality from both code and memory forensics perspectives. The tools introduced in this section include CFF Explorer, Scylla, OllyDumpEx, Volatility, and others.

TOPICS: Recognizing Packed Malware; Getting Started with Unpacking; Using Debuggers for Dumping Packed Malware from Memory; Analyzing Multi-Technology and Fileless Malware; Code Injection and API Hooking; Using Memory Forensics for Malware Analysis

SECTION 5: Examining Self-Defending Malware

Section 5 takes a close look at the techniques that malware authors commonly use to protect malicious software from being examined. You will learn how to recognize and bypass anti-analysis measures designed to slow you down or misdirect you. In the process, you will gain more experience performing static and dynamic analysis of malware that is able to unpack or inject itself into other processes. You will also expand your understanding of how malware authors safeguard the data that they embed inside malicious executables. As with the other topics covered throughout the course, you will be able to experiment with such techniques during hands-on exercises. This section brings together many of the tools covered earlier in the course, including Ghidra and x64dbg. It also introduces FLOSS, ScyllaHide, and others.

TOPICS: How Malware Detects Debuggers and Protects Embedded Data; Unpacking Malicious Software that Employs Process Hollowing; Bypassing the Attempts by Malware to Detect and Evade the Analysis Toolkit; Handling Code Misdirection Techniques, Including SEH and TLS Callbacks; Unpacking Malicious Executable by Anticipating the Packer's Actions

SECTION 6: Malware Analysis Tournament

Section 6 assigns you to the role of a malware analyst working as a member of an incident response or forensics team. You will be presented with a variety of hands-on challenges involving real-world malware in the context of a fun tournament. These challenges further your ability to perform typical malware analysis tasks and offer additional learning opportunities. The challenges are designed to reinforce skills covered in the first five sections of the course, making use of the popular SANS NetWars educational platform. By applying the techniques learned earlier in the course, you will consolidate your knowledge and shore up skill areas where you might need additional practice. Students at live events who score the highest in the malware analysis challenge will be awarded a coveted SANS Lethal Forensicator coin, bestowing on them the title of "REM Master."

TOPICS: Behavioral Malware Analysis; Dynamic Malware Analysis (Using a Debugger); Static Malware Analysis (Using a Disassembler); JavaScript Deobfuscation; PDF Document Analysis; Office Document Analysis; Memory Analysis