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

GREM
Reverse Engineering Malware
giac.org/grem

6 Day Program | 36 CPEs | Laptop Required

You Will Be Able To

- Build an isolated, controlled laboratory environment for analyzing the 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
- Analyze malicious, often obfuscated JavaScript and PowerShell scripts that are often used as part of attack chains
- 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 code injection, C2 interactions, dropper and downloader techniques, 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.
- Analyze .NET malware, which is often obfuscated and attempts to evade detection by using reflective code loading

NICE Framework Work Roles

- Cyber Defense Incident Responder (OPM 531)
- Cyber Crime Investigator (OPM 221)
- Law Enforcement/CounterIntelligence Forensics Analyst (OPM 211)
- Cyber Defense Forensics Analyst (OPM 212)

"This is 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 malware analysis essentials that let you go beyond 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 derive additional insights and indicators of compromise. You will also start mastering dynamic code analysis techniques with the help of a debugger.

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 the analysis of malicious Microsoft Office, RTF, and PDF document files, which are often used as part of the attack chain in mainstream and targeted attacks. You'll learn how to examine macros and other threats that such documents might pose. The course will also teach you how to deobfuscate malicious scripts in the form of JavaScript and PowerShell scripts. You'll also learn how to examine shellcode.

Malware is often obfuscated to hinder analysis efforts, so the course will equip you with the skills to unpack malicious Windows executables. You will learn how to dump such programs from memory or otherwise bypass the packer's protection with the help of a debugger and additional specialized tools. You will also learn how to examine malware that performs code injection and API hooking to conceal its presence on the system or interfere with information flow.

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 "fileless" techniques, sandbox evasion, flow misdirection, debugger detection, and other anti-analysis 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 lab 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 metadata 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 examining malware in your lab—with guidance and explanations from the instructor to reinforce the concepts discussed throughout the day.

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; Exploring Network Interactions of Malware in a Lab for Additional Characteristics

SECTION 2: Reversing Malicious Code

Section 2 focuses on statically 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 decompiler. You will use Ghidra for hands-on exercises in this section. The section begins with an informally overview of key code-reversing concepts and presents an experiment with aspects of primer on essential x86 Intel assembly concepts, such as malware analysis and are looking to instructions, function calls, variables, and jumps. You will formalize and expand your expertise, also learn how to examine common assembly constructs, in this area such as functions, loops, and conditional statements. The material will then build on this foundation and expand practitioners looking to expand understanding to incorporate 64-bit malware.

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; Recognizing Common Malware Characteristics at the Windows API Level; Extending Assembly Knowledge to Include x64 Code Analysis

SECTION 3: Beyond Traditional Executables

Section 3 explores malware samples and techniques that thrive in the Windows ecosystem but that are not traditional executable files. We'll cover the analysis of malicious documents—PDF, Microsoft Office, and RTF files. We will also learn ways of examining the threat posed by suspicious websites. This section will also share techniques for analyzing shellcode, handling JavaScript and PowerShell scripts, and examining .NET malware.

TOPICS: Malicious PDF File Analysis; Malicious Website Analysis; VBA Macros in Microsoft Office Documents; Examining Malicious RTF Files; Understanding Shellcode; Deobfuscating Malicious JavaScript Scripts; Analyzing PowerShell and .NET Malware

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 discussing practical methods for deobfuscating JavaScript, which you might encounter in malicious documents, suspicious websites, and other forms of attacks. Next, you'll learn how to handle packed malware. You will explore ways to identify packers and strip away their protection with the help of a debugger and other utilities. You will also examine a malware sample that employs multiple technologies to conceal its true nature, including the use of registry, obfuscated JavaScript and PowerShell scripts, and shellcode. You will also learn how to analyze .NET malware that has been obfuscated or packed. Finally, you will learn how malware performs code injection to evade detection and interfere with how programs perceive their environment.

TOPICS: Recognizing Packed Windows Malware; Getting Started with Unpacking; Using Debuggers for Dumping Packed Malware from Memory; Analyzing Multi-technology and "Fileless" Malware; Examining .NET Malware (in-depth); Code Injection Techniques

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

GREM
Reverse Engineering Malware
giac.org/grem

GIAC Reverse Engineering Malware

The GIAC Reverse Engineering Malware (GREM) certification is designed for technologists who protect the organization from malicious code. GREM-certified technologists possess the knowledge and skills to reverse-engineer malicious software (malware) that targets common platforms, such as Microsoft Windows and web browsers. These individuals know how to examine inner-workings of malware in the context of forensic investigations, incident response, and Windows system administration. Become more valuable to your employer and/or customers by highlighting your cutting-edge malware analysis skills through the GREM certification.

- Malware Analysis Using Malware Code and Behavioral Analysis Fundamentals
- Windows Assembly Code Concepts for Reverse Engineering and Common Windows Malware Characteristics in Assembly
- In-Depth Analysis of Malicious Executables and Self-Defending Malware
- Analysis of Malicious Document Files, .NET programs, and Protected Executables

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 analyzed. 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 and expands on many of the tools and techniques covered earlier in the course.

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 Tools; Handling Code Misdirection Techniques, Including SEH and TLS Callbacks; Unpacking Malicious Executables by Anticipating the Packer's Actions

SECTION 6: Malware Analysis Tournament

Section 6 allows you to internalize, practice, and expand the many aspects of malware analysis you learned in the earlier sections of the course. You will be presented with a variety of hands-on challenges involving real-world malware in the context of a fun tournament. You will be given access to a capture-the-flag (CTF) system that will present to you practical challenges, which you'll need to address by examining malware in your lab. The system will offer guidance when you need it, so you can cater this game experience to your own skillset and needs. The tournament will help you consolidate your knowledge and shore up skill areas where you might need additional practice.

TOPICS: Malware Analysis Fundamentals; Reversing Malicious Code Using Static and Dynamic Techniques; Analyzing Malicious Documents; In-depth Malware Analysis, Including Unpacking; Examining Self-defending Malware