

FOR610: Reverse-Engineering Malware: Malware Analysis in the Age of AI™



GREM
Reverse Engineering
Malware
giac.org/grem

6
Day Program

36
CPEs

Laptop
Required

You Will Be Able To

- Set up a secure lab and monitor how malware interacts with its environment
- Direct AI tools through an investigation, verify their conclusions, and fill in their gaps with your own analysis
- Analyze obfuscated scripts, executables, and other common infection artifacts
- Control malware through network interception and code patching
- Investigate internals using disassemblers, decompilers, and debuggers
- Identify assembly-level patterns like code injection and anti-analysis tactics
- Extract IOCs and assess threats from malicious files

Business Takeaways

- Enable teams to perform in-house malware analysis, reducing reliance on external resources
- Expand analysis capabilities to provide greater value to internal or external stakeholders
- Increase efficiency in delivering actionable insights and analysis results
- Minimize incident scope and cost with faster response times to security threats
- Meet the growing enterprise demand for analysts who use AI to reach senior-level depth

“This course has helped me to improve my knowledge of malware techniques. I’ve taken nine SANS courses in three years, and FOR610 is one of the best courses SANS offers. Even if malware reversing isn’t part of one’s regular duties, it’s still a worthwhile class just for exposure and enjoyment.”

—Chris A., U.S. Military

Thousands of professionals who analyze malware have gotten their start by taking FOR610: Reverse-Engineering Malware. This practical course is the onramp into this fun and challenging aspect of cybersecurity. It will teach you to dissect malicious software targeting Windows systems by combining expert instruction and immersive hands-on labs.

You’ll learn how to go deeper than ever before with AI tooling. You’ll direct AI to assist with your investigation, validate its findings, fill in its gaps, and arrive at your own conclusions.

The training moves beyond automated tools, showing you how to use disassemblers, debuggers, and monitoring utilities to interpret behavioral patterns, extract indicators of compromise, and gain key insights into malware operations. In addition to focusing on compiled Windows executables, the course covers the analysis of malicious documents, scripts, shellcode, and other artifacts common in modern infections. It also prepares you to counter anti-analysis techniques, such as sandbox evasion and anti-debugging.

With engaging labs and capture-the-flag challenges, FOR610 lets you expand your skills by tackling real-world incident response and forensics investigations that involve malicious software.

In summary, FOR610 malware analysis training will teach you how to:

- Build an isolated, controlled lab environment to safely analyze the code and behavior of malicious programs
- Direct AI tools through a malware investigation and validate their findings in your lab
- Use network and system-monitoring tools to investigate malware interactions with files, the registry, networks, and processes in Windows environments
- Analyze obfuscated JavaScript and PowerShell scripts often used in attack chains.
- Intercept network traffic and patch code to control malware behavior during analysis
- Employ disassemblers and debuggers to examine the internal workings of malicious executables
- Bypass packers and anti-analysis mechanisms designed to hinder and misdirect analysts
- Identify common assembly-level patterns like code injection, C2 interactions, and anti-analysis measures
- Assess threats from malicious PDF documents and the delivery chains behind them
- Extract Indicators of Compromise (IOCs) from malware to enhance threat intelligence and incident response efforts
- Reverse-engineer .NET malware designed to evade detection through obfuscation and reflective code loading
- Assess the accuracy of AI-derived malware analysis findings by tracing and testing their consequential claims

“I learned a great amount of valuable information in FOR610, including what areas I need to master for my job. The CTF lab was a wakeup call regarding how much I don’t know, so thank you!”

—Urban M., CNF Technologies

Section Descriptions

SECTION 1: Malware Analysis Fundamentals

In Section 1, you will learn the malware analysis workflow you will use throughout the course. You'll start by directing an AI agent to examine a malware sample and draft an initial report. You'll then verify its claims in your lab, from static properties to behavior to a first debugger session, going deeper than AI could go on its own.

TOPICS:

- Assembling an AI-enabled toolkit for malware analysis
- Examining static properties of suspicious programs
- Performing behavioral analysis of malicious Windows executables
- Directing an AI agent through analysis and validating its findings
- Using emulation and a debugger to reveal what a sample conceals

SECTION 3: Beyond Traditional Executables

Section 3 covers the malware you read rather than disassemble. You will analyze malicious documents and websites, deobfuscate JavaScript and PowerShell, follow a multi-stage delivery chain, and decompile .NET payloads. AI tools will deobfuscate and explain some of the artifacts, and you'll review and build upon their findings with independent methods in your lab.

TOPICS:

- Triaging malicious PDF document lures and delivery chains
- Malicious website analysis
- Following multi-stage infection chains
- Verifying AI-assisted deobfuscation with independent methods
- Analyzing multi-artifact infection chains from lure to payload
- Deobfuscating malicious JavaScript scripts
- Analyzing PowerShell and .NET malware

SECTION 5: Examining Self-Defending Malware

Section 5 covers the ways malware resists analysis. You will recognize and bypass anti-debugging checks, environment checks, and code misdirection such as exception handling and TLS callbacks. You will then recover code that exists in readable form only while it runs. Along the way, an AI agent will help you turn a debugger procedure into a script that automates the analysis process.

TOPICS:

- How malware detects debuggers and protects embedded data
- Using hardware breakpoints and exception settings against resistant malware
- Bypassing the attempts by malware to detect and evade analysis tools
- Handling code misdirection through exception handling and TLS callbacks
- Recovering code that a sample decodes only while it runs

SECTION 2: Reversing Malicious Code

Section 2 dives into assembly-level analysis of Windows executables, teaching key concepts for code-level analysis using tools such as Ghidra. You'll learn to interpret disassembled code, follow control flow, and identify common malware characteristics, such as command and control, through hands-on exercises. Critically, you'll learn how to speed up and deepen your code analysis using AI tooling.

TOPICS:

- Understanding core assembly concepts for malicious code analysis
- Identifying key assembly constructs with a disassembler
- Following program control flow to understand decision points
- Identifying how malware resolves the Windows APIs it calls at runtime
- Extending assembly knowledge to cover 32-bit and 64-bit code analysis

SECTION 4: Unpacking and In-Memory Execution

Section 4 covers native code that becomes visible only as the malware runs. You will unpack protected malware, extract and debug shellcode, and follow how malware injects code into other processes. You will learn the common sequence behind process injection, so you can recognize any variation. The AI agent will assist with unpacking, and you'll confirm the results in the debugger.

TOPICS:

- Recognizing packed Windows malware
- Getting started with unpacking
- Using debuggers for dumping packed malware from memory
- Analyzing multi-technology and "fileless" malware
- Extracting and debugging shellcode
- Code injection techniques, including process hollowing

SECTION 6: Malware Analysis Tournament

Section 6 consolidates malware analysis skills through a capture-the-flag tournament, offering practical challenges with real-world malware. This hands-on experience reinforces key techniques in static and dynamic analysis, unpacking, and examining self-defending malware.

TOPICS:

- Malware analysis fundamentals
- Reversing malicious code using static and dynamic techniques
- Analyzing scripts, managed code, and malware delivery chains
- Unpacking and in-memory execution, including code injection
- Examining self-defending malware

Who Should Attend

- Practitioners managing incidents involving malware who need to understand its behavior and impact
- Technologists with informal malware analysis experience looking to formalize and expand their expertise
- Forensic investigators and IT professionals aiming to strengthen their role in the incident response process
- Professionals in roles such as incident response, digital forensics, threat intelligence, or SOC operations who want to deepen their reverse engineering and dynamic analysis capabilities

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)

What is Reverse Engineering Malware And Why Is It Important?

Reverse engineering malware means examining malicious software to determine what it does and what threat it poses. This course uses that term and malware analysis interchangeably. Malware is part of many cybersecurity incidents. Knowing how to analyze it lets you take control of the incident and speed up your response.



GREM

Reverse Engineering Malware
giac.org/grem

GIAC Reverse Engineering Malware

The GIAC Reverse Engineering Malware (GREM) certification validates a practitioner's ability to examine the inner workings of malware in the context of forensic investigations, incident response, and Windows system administration. It proves the skills to reverse-engineer malicious software targeting common platforms, including Microsoft Windows and web browsers.