

Basic Malware Analysis Using Responder™ Professional (3-day Instructor-led Course)

CPE Credits: 24

Level: Intermediate

Prerequisites:

- Intermediate computer skills (At least 3 years experience using the Windows operating system)
- At least 2 years of computer forensics experience, or 1 year of network intrusion investigation experience
- No prior experience in software reverse engineering is necessary, but is recommended

Who should attend?

- Owners of HBGary Responder who want to increase their effectiveness with the tool
- System administrators and incident-handling personnel who are trying to further their knowledge in the latest malware detection and analysis techniques
- Anyone who wants to learn how to detect and analyze malware on live Windows systems

Course Objectives:

By the end of the course, students will be able to:

- Utilize methods for preserving live memory and analyzing memory snapshots
- Identify current trends in malicious attacks and how HBGary Responder™ is adapting to address them
- Describe Microsoft Windows Operating system internals and memory features
- List HBGary reverse engineering levels and requirements
- Edit the Baserules.txt file
- Identify, and describe Windows API functionality
- Identify, diagnose and triage malware
- Utilize methods to search memory heaps and stacks for evidentiary artifacts
- Identify malware anti-detection techniques

Course Outline:

Day 1

- Introductions
- Difficulty levels of reverse engineering (I – IV)
 - I – Recovery of a single string/symbol.
 - II – Requires only a single point RE of an API call
 - III – Requires RE of a set of functions and branches
 - IV – Algorithm reconstruction & programming skills
- Role of Physical Memory in Incident Response
- Windows O/S Layout and Internals
- Introduction to HBGary Responder Professional interface and panels
- Introduction to Malware Threat
- Common Malware Behavior
- DDNA Panel
- Introduction to API Calls
- Directories, Files and Downloads Lecture/Hands-on Lab

Day 2

- Registry keys Lecture/Hands-on Lab
- How to reconstruct arguments to an API Call Lecture/Hands-on Lab
- Format Strings Lecture/Hands-on Lab
- Droppers and Multistage Execution Lecture/Hands-on Lab
- Keylogging, Passwords and Datatheft Lecture/Hands-on Lab
- Shell Extensions Lecture/Hands-on Lab

Day 3

- Browser Extensions Lecture/Hands-on Lab
- DLL and Thread Injection Lecture/Hands-on Lab
- REcon Lecture/Hands-on Lab
- “Capture the Flag” team lab competition