MalOpSec 2 -> EDR: The Great Escape

Introduction

Engaging in red-team activities within enterprise networks often involves encountering and bypassing endpoint protection solutions, specifically Endpoint Detection and Response (EDR) systems. These EDRs are intricate and sophisticated systems designed to monitor and defend against various threats, including unauthorized access attempts by red team operators seeking to infiltrate the target network.

This course aims to provide a comprehensive understanding of the architecture of modern EDRs and their underlying Antivirus (AV) systems. It delves deeply into the complexity of modern EDRs, their structure, including the components responsible for real-time monitoring, data collection, and threat analysis.

The course also explores how internal Antivirus (AV) systems operate within the EDR framework, their role in threat detection, and their interaction with other security components.

In addition to examining detection mechanisms employed by EDRs, participants will learn about evasion techniques. This includes tactics and strategies to evade detection by EDRs, such as bypassing signature-based scans, disguising malicious behaviour, and exploiting potential vulnerabilities in EDR configurations.

The techniques will be demonstrated in two ways: first, by reversing real malware samples, and then by re-implementing an improved version of the malware code.

The training is designed from an attacker's point of view, teaching red-teams how to make their implants stealthier, but it will also teach defenders how to deal with the anti-reversing and the OPSEC techniques demonstrated in class.

The course focuses on Windows malware and on the analysis, tweaking and re-purposing of real malware samples. Participants will be provided with plenty of custom code to facilitate the understanding of complex malware techniques.

As part of the course, theory sessions will be followed by exercises where participants will reverse and re-implement specific parts of real malware in order to fully understand the hidden corners of all the techniques involved. The 50% of the course will be dedicated to hands-on labs that will show how to translate the theory principles into practice.

Labs are designed to provide flexibility in terms of complexity and include bonus tracks to ensure that you always feel engaged and have something interesting to explore and learn.

This Class is complementary to our main training covering techniques not present in the main class.

This course is valuable not only for red team operators but also for blue team professionals. Blue team members can gain insights into how their detection systems may be bypassed, helping them enhance their security measures and stay one step ahead of potential threats.

This course equips security professionals with a deep understanding of modern EDRs and their AV systems, enabling them to better simulate advanced threat scenarios, improve their evasion detection skills, and contribute to the overall enhancement of security within enterprise networks.

Key Learning Objective

- **Be able to recognize, implement and deal with stealthy malware/backdoors evasion techniques and tradecrafts.**
- **Be able to modify malware components to protect them against reversing efforts.**
- **Familiarize with the .NET advanced obfuscation system.**
- **Be able to build custom obfuscators and to recognize some pattern left by some obfuscation transforms.**
- **Learn tradecrafts used by attackers to prevent and effectively impair defensive incident responders from analysing their tools, payloads, and backdoors.**

Who should attend

Developers and Reverse engineers who want to understand tradecrafts from a different point of view, red-team members who want to go beyond using third-party implants, and researchers who want to develop anti-detection techniques of real malware/apt.

Prerequisites

- **Programming experience (C, C++, Python, .NET, and PowerShell)**
- **Be *comfortable* with assembly language and Debuggers (IDA pro, WinDBG)**

Hardware/Software Requirements:

Laptop Requirements:

- **Virtualization capable CPU(s)**
- **Minimum 8GB of RAM (for running one guest VM)**
- **Minimum 80 GB free disk space**
- **Host CPU intel (ARM is not supported)**

Software Requirements:

- **Host OS Windows 10 64-bit**
- **Debugging Tools for Windows (Ida Pro, WinDBG). Decompiler recommended.**
- ***SysInternals* Toolsuite**
- **Virtualization Software (VMWare, VirtualBox)**
- **Guest OS Windows 10 64-bit Version 20H2**
- **System Administrator access required on both host and guest Oss**

Course Agenda

Module 1

- The shortest Intro
- Give a shout to the Alpaca
- The reference architecture
- Minifilter drivers
  - Architecture, altitute
  - pre/post operation Callbacks
  - Self-protection
- Kernel to user dll injection
  - APC injection
  - Hooking library
  - Hook detection / Unhooking strategies
  - Look at a couple of proprietary DLL s
- Unhooking the watchers in all the possible ways
  - Restore the original ntdll
  - Patch the hooked ntdll in memory
  - The right ways of using call gates
  - Indirect syscall
- Labs:
  - Unhook
  - Disable self-protection

Module 2

- Using ROP to do good or better bad things…
  - Write your ROP injector
- Protected Processes and Protected Process Light
  - Internals: Core kernel data structures
  - Anti-Malware and ELAM
- Mastering ETW and get the forbidden feed
  - Providers, Consumers, Sessions
  - User-space provider bypass

    - The Threat Intelligence Provider
- Labs:
    - Using ROP to minimize the presence in ETW logs
    - Silence the ETW feed

Module 3

- Memory Scanners
- Smashing the stack for fun and evasions
    - Stack spoofing
    - Sleep Obfuscation
- Local Privilege Escalation and Lateral Movement
    - Windows authentication (Local, Network)
    - NTLM, Kerberos, MS-RPC
    - Abuse WinSxS
    - Handle stealer
- Labs
    - LPE and get Admin
    - Create your stack spoof